Gerimedica Implementation of System for Cross-domain Identity Management (SCIM)

Abstract

This document details the Gerimedica implementation of ‘System for Cross-domain Identity Management’ or SCIM. The intended purpose of this document is to describe the details of our implementation including what aspects of the standards we support and how this system can be used. The intended audience for this document is developers who would like to use SCIM for managing Gerimedica users and their location and ward assignments.

For further information or questions regarding the content of this document, please contact support@gerimedica.nl.

Version

Document Version	Release Date
1.0	27-02-2023
1.01	14-03-2023
1.02	24-03-2023
1.03	19-04-2023
1.04	20-06-2023
1.05	27-07-2023
1.06	09-08-2023
1.07	11-08-2023
1.08	23-01-2024
1.09	19-09-2024
1.10	02-04-2025
1.11	10-09-2025
1.12	16-09-2025

1.13	23-09-2025
1.14	24-09-2025
1.15	19-06-2026
1.16	19-10-2026
1.17	06-11-2026

Changelog

Notable changes to this document will be tracked here.

Version 1.01

- We have updated the list of possible entitlements to now include Module Authorizations. As a result, users can now be granted access to this specific feature or functionality. Please note that this is a new addition and as such, the following values have been added to the list of possible entitlements:
-

```
"AdministratieveOndersteuning",
"Behandeldossier",
"FinancieelVoorGRZBasisGGZELVZPM",
"FinancieelVoorEerstelijnsParamedischeZorgHuisartsenzorgGZSP",
"Gebruikersbeheer",
"Inzicht",
"PortaalFinanciëleKoppeling",
"Zorgdossier"
```

- We have made updates to the request and response of the 'create' and 'update' endpoints, as well as the response of the 'get' function, in accordance with the YSIS extension schema. As a result, the format and structure of these endpoints have been modified to align with the schema. Please take note that only the response of the 'get' function has been changed, while the request and response of the 'create' and 'update' functions have been updated.

So the current usage is like this:

```
ez  "urn:ietf:params:scim:schemas:extension:ysis:2.0:User": {
   "ysisInitials": "Y.S",
   "discipline": "MEDICAL",
```

```
   "agbCode": "agbCode1"
 }
```

We are currently not supporting custom roles. As such, it is not possible to create or assign custom roles to users at this time.
We have removed the 'honorific prefix' field from the user schema. As a result, this field will no longer be displayed or required when creating or updating user information.

Version 1.02

We have added ‘initials’ as a new property to the YSIS User extension Schema. This value is required, but unlike ‘ysisInitials’ does not need to be unique across the organization.

```
  "urn:ietf:params:scim:schemas:extension:ysis:2.0:User": {
   "ysisInitials": "Y.S",
   "discipline": "MEDICAL",
   "agbCode": "agbCode1",
   "initials": "YS"
 }
```

Version 1.03

- We have added a new schema for User Enterprise.

  This User Enterprise schema has an attribute employeeNumber.

```
{
  "id": "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User",
  "name": "Enterprise",
  "description": "Enterprise for User ResourceType.",
  "attributes": [
  {
    "name": "employeeNumber",
    "type": "string",
    "multiValued": false,
    "description": "The employer's number of the user. The maximum size for this field is 20
character.",
    "required": false,
    "caseExact": true,
    "mutability": "readOnly",
    "returned": "default",
    "uniqueness": "server"
  }
  ],
  "meta": {
```

```
    "resourceType": "Schema",
    "location":
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
  }
}
```

- We have added gender, export timeline events and infix to User Schema.

```
 {
        "name": "gender",
        "type": "complex",
        "multiValued": false,
        "description":"An indication of the User's gender. If no one selected
it will be given as UNKNOWN by default.",
        "required": false,
        "canonicalValues": [
          "UNKNOWN",
          "MALE",
          "FEMALE"
        ],
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "exportTimelineEvents",
        "type": "boolean",
        "multiValued": false,
        "description": "A Boolean value indicating whether to show the status
of the User's events on calendar or not.",
        "required": false,
        "caseExact": true,
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "none"
      }
```

- We have added **infix** as subAttribute to Name attribute in user schema

```
    {
        "name": "name",
        "type": "complex",
        "multiValued": false,
        "description": "The components of the user's real name.",
        "required": false,
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "none"
       "subAttributes": [
          {
       "name": "infix",
       "type": "string",
       "multiValued": false,
       "description": "It is a Tussenvoegsel. In a family name affix
positioned between a person's given name and the main part of their family
name. For Example for name: Emma van der Sluss, van der is infix.",
       "required": false,
       "caseExact": true,
       "mutability": "readWrite",
       "returned": "default",
       "uniqueness": "none"
}]}
```

- We have added bignumber and position to **Ysis User Extension Schema**

```
 {
        "name": "bigNumber",
        "type": "string",
        "multiValued": false,
        "description": "Bignumber of the User. The maximum size for this
```

```
field is 11 characters.",
        "required": false,
        "caseExact": true,
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "server"
      },
      {
         "name": "position",
          "type": "string",
          "multiValued": false,
          "description": "The function of the user. The maximum size for this
field is 255 character.",
          "required": false,
          "caseExact": true,
          "mutability": "readWrite",
          "returned": "default",
           "uniqueness": "none"
      }
```

- To create user request new attributes added

```
 "gender": null,
 "exportTimelineEvents": true,
 "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
   "employeeNumber": "2015010213058"
 }
 "name": {
 "infix": "van"
 }
```

- To create user response updated

```
"gender": "UNKNOWN",
"exportTimelineEvents": true,
"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
  "employeeNumber": "2015010213058"
 }
"name": {
 "infix": "van"
}
```

- Update User Request updated

```
"gender": null,
"exportTimelineEvents": true
"name": {
 "infix": "van"
}
```

- Update User Response updated

```
"gender": null,
"exportTimelineEvents": true
"name": {
 "infix": "van"
}
```

- We also added this new schema for all request which are contain schemas json

```
"schemas": [
 "urn:ietf:params:scim:schemas:core:2.0:User",
 "urn:ietf:params:scim:schemas:extension:ysis:2.0:User",
 "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
]
```

Version 1.04

We have made an enhancement to the discipline field by updating the accepted values. In the previous version, we allowed descriptions of disciplines, but now we only accept enum values for the disciplines. To provide more clarity, here are the new enum values and their corresponding mappings:

Discipline Value	**Description**
MEDICAL	Medisch
PHYSIOTHERAPIST	Fysiotherapie
PSYCHOLOGIST	Psychologie
ERGOTHERAPIST	Ergotherapie
LOGOPEDIST	Logopedie
DIETITIAN	Diëtetiek
PASTOR	Geestelijke verzorging
SOCIAL_WORK	Maatschappelijk werk
MUSIC_THERAPIST	Muziektherapie
PMT	Psychomotorische therapie
DENTAL_CARE	Mondzorg
ACTIVITY_COACH	Activiteitenbegeleiding
READ_ONLY	Leesaccount
CARE	Zorg
NURSE	Verpleegkunde
ADMINISTRATIVE	Administratie
PSYCHOLOGIST_CONSULTANT	Psychologie consultatie

Please note that this change impacts the create, update, and get endpoints. If you need to verify the current values, you can refer to the schema's endpoint for the most up-to-date information.

- We have made another improvement regarding the field "roles." From now on, two-worded roles should be provided in regular case. This change applies to the create and update endpoints. Furthermore, the response from the get request will include these roles in the specified format. For your reference, here are the affected values and their updated formats resulting from this change:
    - GeestelijkeVerzorging -> Geestelijke verzorging
    - MaatschappelijkWerk -> Maatschappelijk werk
    - PsychomotorischeTherapie -> Psychomotorische therapie
    - PsychologieConsultatie -> Psychologie consultatie

  To ensure you have the most accurate and current information, you can also check the schema's endpoint for the latest values.

Version 1.05

- We have introduced the ability to explicitly specify which Ysis Modules the user has access to. This includes adding a new non-mandatory ‘modules’ field to the `urn:ietf:params:scim:schemas:extension:ysis:2.0:User` schema. The valid set of possible values for this field are:
    - YSIS_CORE
    - YSIS_DBC
    - ECD_CARE
    - FINANCIAL
    - FINANCIAL_EXPORT_PORTAL
    - MANAGEMENT
    - USER_MANAGEMENT
- The following values were previously valid to pass in as authorization ids. These are no longer valid and passing them in will return a validation error. With the exception of ‘Inzicht’ which has been removed as it is not in use, these values have been migrated to modules:
    - AdministratieveOndersteuning
    - Behandeldossier
    - FinancieelVoorGRZBasisGGZELVZPM
    - FinancieelVoorEerstelijnsParamedischeZorgHuisartsenzorgGZSP
    - Gebruikersbeheer
    - Inzicht
    - PortaalFinanciëleKoppeling
    - Zorgdossier

Version 1.06

- We have updated the User (`urn:ietf:params:scim:schemas:extension:ysis:2.0:User`) schema ‘isActive’ field

to be mutable. This affects the SCIM User create and update endpoints which have been enhanced to process the value of this field and will internally update the ‘blocked’ status of the account. Setting the value to false will block the account, so it cannot be used until it is unblocked, Setting the active status to false does not delete the account. Blocked accounts can be unblocked and made active again by setting the ‘active‘ field to true. **Please note, accounts which are deleted are not set to blocked, they are internally archived and cannot be unblocked by setting the ‘active’ field to true. An archived account can be re-activated by creating a user with the same userName, ysisInitials and discipline, see Creating Users.**

- We have created a new ‘/users’ endpoint, which can return sets of users. This endpoint is not intended to return all available users. The purpose of this endpoint is to determine if a non deleted User exists which matches a given filter. A filter must be provided for this endpoint. Full details of the supported filters can be found in the Filtering section. **The intended use for this functionality is to allow clients to verify that an account does or does not already exist for a given employeeNumber.**
- Filtering is now supported, and the OverviewSchema has been updated to reflect this. Please note, currently filtering is only supported on the new ‘/users’ endpoint.

Version 1.07

We have added a new GET endpoint which can be used to return the available set of Roles for an organization. This will include both the default and custom roles. The ScimRole objects returned from this endpoint are the same object which is currently used in the existing SCIM User endpoints. Please see the SCIM Role Operations section for more information.

We have also modified the existing ScimRole object, we have added a displayName field.

Previous Definition:
Example Error:

```
ScimRole
{
   "value": "The value for this Role."
}
```

New Definition:

```
ScimRole
{
   "value": "The id for this Role.",
   "displayName": "The human readable name of this Role."
}
```

Version 1.08

This document has been updated with more error examples and some minor corrections after a review of the SCIM error messaging.

Version 1.09

The set of available modules has been updated:

MANAGEMENT has been removed.
ECD_ADMIN has been added.

The set of available Entitlements have been updated to reflect changes in the available Gerimedica Authorizations.

The following Entitlements have been added to the available options:
AccorderenWlz,
AccorderenWmo,
DeclarerenWlz,
DeclarerenWmo

Version 1.10

We have updated **retrieving users list with a filter** endpoint; the user schema filter on field EmployeeNumber accepts the PR operator too in addition to EQ.

Version 1.11

We have updated **retrieving users list with a filter** endpoint; the user schema filter accepts on field userName the EQ and PR operators.

Version 1.12

We have created the version v3 for our endpoints because any query resource endpoint will return an object like this:

```
{
   "schemas": [
       "urn:ietf:params:scim:api:messages:2.0:ListResponse"
   ],
```

```
   "totalResults": 7,
   "Resources": [
      ...
   ]
}
```

The affected endpoints are:

- List resource types
- List schemas
- List users
- List roles

Version 1.13

Password is no longer mandatory to create a user. If the password is not provided, there will be an auto generated password.

Version 1.14

Updating schema that wrongly mentioned the roles could be assigned automatically. Our SCIM provisioning does not allow that.

Version 1.15

Updating the authentication method for our acceptance environment migrating to Auth0. Production is scheduled to migrate and this document will have a new version.

Version 1.16

- We have added location and ward assignment through SCIM Groups. Every ward is available as two Groups, one for linked users and one for authorized users. See Scim Group Operations.
- The User resource has a new read-only attribute ‘groups’ which lists the Groups of the user.
- The Group resource type and schema have been added to the discovery endpoints. The service provider configuration now reports that PATCH is supported.
- The list users endpoint has a new query parameter ‘includeArchived’. Archived users are returned as inactive users, so their details can be found and used for re-activation.
- The list users endpoint for v3 no longer requires a filter. Without a filter, users are returned in pages of at most 100. The parameters ‘startIndex’ and ‘count’ select the page. The parameters ‘attributes’ and ‘excludedAttributes’ select the returned attributes.
- Re-activation of archived users is now described. The statement in version 1.06 that archived accounts cannot be re-activated has been corrected. See Creating Users.
- An unsupported filter on the user and Group endpoints now returns 400 Bad Request with scimType ‘invalidFilter’. This was 403 Forbidden.
- A User can now be changed with a PATCH, see Updating Users (PATCH).
- A step by step description of how to assign users to locations and wards has been added, see Assigning Users to Wards.

- Corrections to filtering: the schema id can be left out of a filter, and an empty filter value returns an error instead of no results.
- Production now uses Auth0 for authentication. The $AUTH_URL of Production is **https://auth.ysis.nl/**

Version 1.17

- The list users endpoint supports filtering on ‘ysisInitials’ of the schema urn:ietf:params:scim:schemas:extension:ysis:2.0:User, with the ‘eq’ operator. This can be used to check whether Ysis initials are already in use before creating a user. See Filtering.
- The ‘ysisInitials’ attribute is now described as ‘caseExact’: false in the schema. Ysis initials have always been compared without regard to case.

Abstract **1**
Version **1**
Changelog **2**
Version 1.01 2
Version 1.02 3
Version 1.03 3
Version 1.04 8
Version 1.05 9
Version 1.06 9
Version 1.07 10
Version 1.08 11
Version 1.09 11
Version 1.10 11
Version 1.11 11
Version 1.12 11
Version 1.13 12
Version 1.14 12
Version 1.15 12
Version 1.16 **12**
Version 1.17 13
SCIM **17**
What Is SCIM 17
Accessing SCIM **18**
Authentication **19**
Access Token 19
How To Get an Access Token 19
Example Request 19
Example Response 19
Example Error Response 20
How To Use Access Token 20
Ip Whitelist 20
Gerimedica SCIM Implementation **22**
Service Provider Configuration Schema 22
Example Successful Response 22
Example Error Responses 23
Resource Types Schemas 25
Resource Types 25
Example Successful Response 25
Example Error Responses 26
Resource Type 28
Example Successful Response 28
Example Error Responses 28
Schema Representations 31

Schema Definitions 31
Example Successful Response 31
Example Error Responses 59
Schema Definition 61
Example Successful Response - User 61
Example Successful Response - User Enterprise Extension 70
Example Successful Response - Ysis User Extension 71
Example Successful Response - Group 73
Example Successful Response - ResourceType 76
Example Successful Response - ServiceProviderConfig 79
Example Successful Response - Schema 85
Example Successful Response - Error 94
Example Error Responses 95
Scim User Operations 97
Creating Users 97
Example Request 97
Example Successful Response 98
Example Error Responses 100
Re-activating Archived Users 101
Retrieving Users (by userId) 103
Example Request 103
Example Successful Response 103
Example Error Responses 105
Retrieving Users (with Filter) 107
Example Request 107
Example Successful Response 107
Example Error Responses 109
Including Archived Users 110
Example Request 110
Example Successful Response 110
Filtering 112
Supported Filter Structure: 112
Supported SchemaIds and AttributeName: 112
Supported Operators: 112
Supported Values: 112
Updating Users (PUT) 113
Example Request 113
Example Successful Response 114
Example Error Response 114
Updating Users (PATCH) 117
Example Request 117
Example Successful Response 118
Example Error Responses 119
Deleting Users 122

Example Request 122
Example Successful Response 122
Example Error Responses 122
Scim Role Operations 124
Retrieving Roles 124
Example Request 124
Example Successful Response 124
Example Error Responses 125
Scim Group Operations 126
Locations and Wards as Groups 126
Assigning Users to Wards 126
Asynchronous Processing 127
Retrieving Groups (List) 128
Example Request 128
Example Successful Response 128
Example Error Responses 129
Retrieving Groups (by groupId) 129
Example Request 130
Example Successful Response 130
Example Error Responses 130
Updating Group Members (PATCH) 131
Example Request 132
Example Successful Response 133
Example Error Responses 133
Replacing Group Members (PUT) 135
Example Request 135
Example Successful Response 135
Example Error Responses 136
Creating and Deleting Groups 136
Groups of a User 137
Example 137
Frequently Asked Questions 138

SCIM

What Is SCIM

SCIM stands for System for Cross-domain Identity Management.

It is an open source protocol for automating the exchange of user identity information across domains. It provides a standardized way for different systems to share user profiles and groups and avoids the duplicated work of creating the same users in multiple systems manually. SCIM allows our customers to automate and streamline the management of Gerimedica users resulting in a more secure and efficient process.

Gerimedica implements Version 2.0 of SCIM.

Technical information can be found for the SCIM protocol here:

Overview: https://www.rfc-editor.org/rfc/rfc7642
Core-Schema: https://www.rfc-editor.org/rfc/rfc7643
Protocol: https://www.rfc-editor.org/rfc/rfc7644

Accessing SCIM

All SCIM relative endpoints presented in this document are relative to a base URL, which includes a subdomain that is unique per organization who is a customer of Gerimedica. Subdomains will be displayed as **{subdomain}**, customers should substitute this with your own unique subdomain value.

SCIM is available on both our Acceptance and Production environments. Relative SCIM endpoints will be presented in the rest of the document as relative to the **$BASE_URL**.

The **$BASE_URL** per environment:
Production: **https://{subdomain}.ysis.nl/gm/api/um/**
Acceptance: **https://{subdomain}.acceptatie2.ysis.nl/gm/api/um/**

Authentication is provided by Auth0. Our authentication endpoints will be presented relative to the **$AUTH_URL**.

The **$AUTH_URL** per environment:
Production: **https://auth.ysis.nl/**
Acceptance: **https://auth.acceptatie2.ysis.nl/**

Authentication

To reach any SCIM endpoint, the caller is required to have an Oauth2 access token. This Access Token must be sent in the header of all requests in the form of a bearer token. Anonymous SCIM requests are not supported.

Access Token

How To Get an Access Token

In order to get a valid Access Token, callers are required to provide a **client id** and **client secret**. If you do not have this information, please contact Gerimedica support via support@gerimedica.nl. Access Tokens have an expiration time of 30 minutes, at which point the caller will need to get a new Access Token.

URL: **$AUTH_URL/oauth/token**
HTTP Operation: **POST**

Example Request

```
HTTP/1.1 POST $AUTH_URL/oauth/token
Content-Type: application/json
{
 "client_id": "client_id"
 "client_secret": "secret"
 "grant_type": "client_credentials"
 "audience": "gerimedica-ext"
}
```

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json
{
   "access_token":
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJzY2ltLWludGVncmF0aW9uIiwiYXVkIjpbInVtcy1hcGkiXSwic2NvcGUiOlsic2NpbSJdLCJleHAiOjE2NzY2MzA3OTYsImF1dGhvcml0aWVzIjpbIlJPTEVfQ0xJRU5UIl0sImp0aSI6Ijdob2ttZFdvWDRGa2hsZnNtVERjU0g2eVQyayIsImNsaWVudF9pZCI6InNjaW0taW50ZWdyYXRpb24ifQ.o4lcNpuc965_z32h2CaDOYdLAbF_yvrEWP5jQT5YOmafoCVHoojsipvoIY0ILn7sneZAAdLmY2Gill-TWJP-9IsBx5lbKBxuWRi96VOH4r5p8u60IfsQfHXxt8w7ImZnf_8aHdIX_86-Q2yy0Sd2QUOzvmeSFJoGHjIIwJYzJdzyvib-R_K2-W3KDNLXothoR6hHzt7OCQsm2U6R0K_B9PoFPrLmim3F_oitSD4D3efIfc7FS8nFB-08bn2xrgHH_EOSh1CxCxnGSgI994Cxd6emAaGyJ8HTuY0zJ28GWHCTz4Jes97-4mReO7PKNHn-W9elLtHSe44OjSNxHu2503lW_TVvFvpKtcLErhGuOycvpz3fJ7t3SwsJOdswvtRFGh2eDQpiTRT-aGVXPTEdx4XPAjMyQ4Vj1iUW3_GPtq1g4zcEbIl4AqZK921ycNGiz-oDt3g0RLll5hF_iN86X3bL-7lTDkAXECFeK1ezb9a7ZnR2--ornRou1Rg1wtEvYGFaZSTgeDR5fUURZwtVeJO9b4BQPJ1P2hIwZDvwcG2au-XquCPa7-mcI2mfsh15Z_Be5UvwQcCIuoPGXlCShEogeDqGGwsBD2PNO3hG0c3cyii6FcSvPWkfPJWZIHLGJEjNoRlqPZReTf85Aq0vP74sdAOidRnqbrcJHpjvtww",
   "token_type": "bearer",
   "expires_in": 1799,
   "scope": "scim"
```

```
}
```

Example Error Response

Error Scenario: The caller attempts to retrieve a token with an invalid client id or client secret.

Example Error:

```
HTTP/1.1 401 Unauthorized
{
   "error": "invalid_client",
   "error_description": "Bad client credentials"
}
```

Error Scenario: The caller attempts to retrieve a token with an invalid grant type.

Example Error:

```
HTTP/1.1 400 Bad Request
{
   "error": "unsupported_grant_type",
   "error_description": "Unsupported grant type"
}
```

How To Use Access Token

In the header of all requests the user needs to provide the bearer token.

```
Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJzY2ltLWludGVncmF0aW9uIiwiYXVkIjpbInVt
cy1hcGkiXSwic2NvcGUiOlsic2NpbSJdLCJleHAiOjE2NzY2MzA3OTYsImF1dGhvcml0aWVzIjpbIlJPTEVfQ
0xJRU5UIl0sImp0aSI6Ijdob2ttZFdvWDRGa2hsZnNtVERjU0g2eVQyayIsImNsaWVudF9pZCI6InNjaW0taW
50ZWdyYXRpb24ifQ.o4lcNpuc965_z32h2CaDOYdLAbF_yvrEWP5jQT5YOmafoCVHoojsipvoIY0ILn7sneZA
AdLmY2Gill-TWJP-9IsBx5lbKBxuWRi96VOH4r5p8u60IfsQfHXxt8w7ImZnf_8aHdIX_86-Q2yy0Sd2QUOzv
meSFJoGHjIIwJYzJdzyvib-R_K2-W3KDNLXothoR6hHzt7OCQsm2U6R0K_B9PoFPrLmim3F_oitSD4D3efIfc
7FS8nFB-08bn2xrgHH_EOSh1CxCxnGSgI994Cxd6emAaGyJ8HTuY0zJ28GWHCTz4Jes97-4mReO7PKNHn-W9e
lLtHSe44OjSNxHu2503lW_TVvFvpKtcLErhGuOycvpz3fJ7t3SwsJOdswvtRFGh2eDQpiTRT-aGVXPTEdx4XP
AjMyQ4Vj1iUW3_GPtq1g4zcEbIl4AqZK921ycNGiz-oDt3g0RLll5hF_iN86X3bL-7lTDkAXECFeK1ezb9a7Z
nR2--ornRou1Rg1wtEvYGFaZSTgeDR5fUURZwtVeJO9b4BQPJ1P2hIwZDvwcG2au-XquCPa7-mcI2mfshl5Z_
Be5UvwQcCIuoPGXlCShEogeDqGGwsBD2PNO3hG0c3cyii6FcSvPWkfPJWZIHLGJEjNoRlqPZReTf85Aq0vP74
sdAOidRnqbrcJHpjvtww
```

Ip Whitelist

To help protect access to our access token endpoint, Gerimedica uses IP whitelisting. This means that we only allow requests to the access token endpoint from specific IP addresses.

If you are a customer and need to call the access token endpoint, you will need to provide Gerimedica with a list of your public IP addresses. We will then add those IP addresses to our whitelist, so that your requests can be authenticated.

To provide us with your list of public IP addresses, please contact our support team at support@gerimedica.nl. Once we have received your list, we will add your IP addresses to our whitelist as soon as possible.

Please note that if your IP address changes or if you need to use a different IP address to make requests to the access token endpoint, you will need to notify us so that we can update our whitelist accordingly.

Gerimedica SCIM Implementation

Service Provider Configuration Schema

Endpoint: **$BASE_URL/scim/v3/service-provider-config**
HTTP Operation: **GET**

This endpoint will return a JSON object which describes the SCIM features supported by Gerimedica. The returned object will match the schema **urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig**.

This endpoint is read-only.

Filtering, sorting, and pagination are not supported.

Example Successful Response

```
HTTP/1.1 200 OK
Content-Type: application/json
{
    "schemas": ["urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig"],
    "documentationUri": "https://apihelp.gerimedica.nl/category/scim/",
    "patch": {
      "supported": true
    },
    "bulk": {
      "supported": false
    },
    "filter": {
      "supported": true
    },
    "changePassword": {
      "supported": false
    },
    "sort": {
      "supported": false
    },
    "etag": {
      "supported": false
    },
    "authenticationSchemes": [
      {
        "name": "OAuth Bearer Token",
        "description":
          "Authentication scheme using the OAuth Bearer Token Standard",
        "specUri": "http://www.rfc-editor.org/info/rfc6750",
        "documentationUri": "http://example.com/help/oauth.html",
        "type": "oauthbearertoken",
        "primary": true
```

```
      }
    ],
    "meta": {
      "location": "$BASE_URL/scim/v3/service-provider-config",
      "resourceType": "ServiceProviderConfig",
      "created": "2023-01-01T04:56:22Z",
      "lastModified": "2023-02-13T04:42:34Z",
      "version": "W/\"037fbfb28473d1590ba2211690684a3de\""
    }
  }
```

Example Error Responses

Error Scenario: A request to this endpoint which contains filtering options will result in an error.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "scimType": "invalidFilter",
  "detail": "Filtering is not supported for this operation."
}
```

Error Scenario: A request to this endpoint is made which does not contain the correct authentication.
Example Error:

```
HTTP/1.1 403 UNAUTHORIZED
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "detail": "Operation is not permitted based on the supplied authorization.",
  "status": 401
}
```

Error Scenario: A request is made to this endpoint from an organization with multiple groups.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations with multiple groups."
}
```

Error Scenario: A request is made to this endpoint from an organization with no group.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
```

```
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations without groups."
}
```

Resource Types Schemas

Resource Types

Endpoint: **$BASE_URL/scim/v3/resource-types**
HTTP Operation: **GET**

A HTTP GET call to this endpoint can be used to return a list of Resource Types which are supported by Gerimedica. Gerimedica currently supports the ‘User’ and ‘Group’ Resource Types and does not support the ‘Me’ Resource Type.

The data returned by this endpoint will be a ListResponse containing Resource Type objects matching the **urn:ietf:params:scim:schemas:core:2.0:ResourceType** schema.

This endpoint is read only.

Filtering, sorting and pagination are not supported.

Example Successful Response

```
HTTP/1.1 200 OK
Content-Type: application/json
{
   "schemas": [
       "urn:ietf:params:scim:api:messages:2.0:ListResponse"
   ],
   "totalResults": 2,
   "Resources": [
       {
         "schemas": ["urn:ietf:params:scim:schemas:core:2.0:ResourceType"],
         "id": "user",
         "name": "User",
         "endpoint": "/user",
         "description": "User Account",
         "schema": "urn:ietf:params:scim:schemas:core:2.0:User",
         "schemaExtensions": [
           {
             "schema":"urn:ietf:params:scim:schemas:extension:ysis:2.0:User",
             "required": true
           },
           {
             "schema": "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User",
             "required": false
           }
         ],
         "meta": {
           "location": "$BASE_URL/scim/v3/resource-types/user",
           "resourceType": "ResourceType"
         }
       },
```

```
      {
        "schemas": ["urn:ietf:params:scim:schemas:core:2.0:ResourceType"],
        "id": "group",
        "name": "Group",
        "endpoint": "/Groups",
        "description": "Ward Group: the users linked to, or authorized on, one ward.
Only the members can be changed",
        "schema": "urn:ietf:params:scim:schemas:core:2.0:Group",
        "schemaExtensions": [],
        "meta": {
          "location": "$BASE_URL/scim/v3/resource-types/group",
          "resourceType": "ResourceType"
        }
      }
    ]
}
```

Example Error Responses

Error Scenario: A request to this endpoint which contains filtering options will result in an error.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": "403",
  "scimType": "invalidFilter",
  "detail": "Filtering is not supported for this operation."
}
```

Error Scenario: A request to this endpoint is made for an invalid organization.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "detail": "Operation is not permitted based on the supplied authorization."
}
```

Error Scenario: A request is made to this endpoint from an organization with multiple groups.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations with multiple groups."
}
```

Error Scenario: A request is made to this endpoint from an organization with no group.

Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations without groups."
}
```

Resource Type

Endpoint: **$BASE_URL/resource-types/{resourceType}**
HTTP Operation: **GET**

A HTTP GET call to this endpoint will return details of a single Resource Type which is supported by Gerimedica. Passing an invalid value for the resourceType parameter will result in an error.

The data returned by this endpoint will be a single JSON object matching the **urn:ietf:params:scim:schemas:core:2.0:ResourceType** schema.

This endpoint is read only.

Filtering, sorting and pagination are not supported.

The following values are valid for the path parameter **resourceType**:

- **User**
- Group

Example Successful Response

```
HTTP/1.1 200 OK
Content-Type: application/json
{
     "schemas": ["urn:ietf:params:scim:schemas:core:2.0:ResourceType"],
     "id": "user",
     "name": "User",
     "endpoint": "/user",
     "description": "User Account",
     "schema": "urn:ietf:params:scim:schemas:core:2.0:User",
     "schemaExtensions": [
       {
         "schema": "urn:ietf:params:scim:schemas:extension:ysis:2.0:User",
         "required": true
       },
       {
         "schema": "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User",
         "required": false
       }
     ],
     "meta": {
       "location": "$BASE_URL/scim/v3/resource-types/user",
       "resourceType": "ResourceType"
     }
}
```

Example Error Responses

Error Scenario: A request to this endpoint which contains filtering options will result in an error.

Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "scimType": "invalidFilter",
  "detail": "Filtering is not supported for this operation."
}
```

Error Scenario: A request to this endpoint which contains an invalid resourceType.
Example Error:

```
HTTP/1.1 400 BAD_REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidValue",
  "detail": "The path parameter 'departments' for 'resourceType' is not supported."
}
```

Error Scenario: A request to this endpoint is made for an invalid organization.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "detail": "Operation is not permitted based on the supplied authorization."
}
```

Error Scenario: A request is made to this endpoint from an organization with multiple groups.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations with multiple groups."
}
```

Error Scenario: A request is made to this endpoint from an organization with no group.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations without groups."
}
```

Schema Representations

Schema Definitions

Endpoint: **$BASE_URL/scim/v3/schemas**
HTTP Operation: **GET**

A HTTP GET to this endpoint can be used to return a complete list of the SCIM Schemas supported by Gerimedica.

The data returned by this endpoint will be in a JSON array format of objects matching the **urn:ietf:params:scim:schemas:core:2.0:Schema** schema.

This endpoint is read only.

Filtering, sorting and pagination are not supported.

Example Successful Response

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 1,
  "Resources": [
  {
    "id": "urn:ietf:params:scim:schemas:core:2.0:User",
    "name": "User",
    "description": "User Account",
    "attributes": [
      {
        "name": "userName",
        "type": "string",
        "multiValued": false,
        "description": "Unique identifier for the User, typically
used by the user to directly authenticate to the service provider.
Each User MUST include a non-empty userName value.  This identifier
MUST be unique across the service provider's entire set of Users.
The maximum size for this field is 40 characters.",
        "required": true,
        "caseExact": false,
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "server"
      },
```

```
      {
        "name": "name",
        "type": "complex",
        "multiValued": false,
        "description": "The components of the user's real name.",
        "required": false,
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "none"
        "subAttributes": [
          {
            "name": "familyName",
            "type": "string",
            "multiValued": false,
            "description": "The family name of the User, or
last name in most Western languages (e.g., 'Jensen' given the full
name 'Ms. Barbara Jensen'). The maximum size for this field is 100
characters.",
            "required": true,
            "caseExact": true,
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "givenName",
            "type": "string",
            "multiValued": false,
            "description": "The given name of the User, or
first name in most Western languages (e.g., 'Barbara' given the
full name 'Ms. Barbara Jensen'). The maximum size for this field is 80
characters.",
            "required": true,
            "caseExact": true,
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "infix",
            "type": "string",
            "multiValued": false,
            "description": "It is a Tussenvoegsel in Dutch. In a family name
affix positioned between a person's given name and the main part of their
family name. For Example for name: Emma van der Sluss, van der is infix.",
",
```

```
            "required": false,
            "caseExact": false,
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          }
        ]
      },
      {
        "name": "active",
        "type": "boolean",
        "multiValued": false,
        "description": "A Boolean value indicating the User's administrative
status.",
        "required": false,
        "mutability": "readWrite",
        "returned": "default"
      },
      {
        "name": "password",
        "type": "string",
        "multiValued": false,
        "description": "The User's cleartext password.  This
attribute is intended to be used as a means to specify an initial
password when creating a new User. A password is OPTIONAL when creating a
user and must not be present otherwise. If the password is not present at
creation, it will be auto generated. A password must be a minimum of 12
characters and a maximum of 64 characters.  All characters are valid except
'/' and '\'.",
        "required": false,
        "caseExact": true,
        "mutability": "writeOnly",
        "returned": "never",
        "uniqueness": "none"
      },
      {
        "name": "emails",
        "type": "complex",
        "multiValued": true,
        "description": "Email addresses for the user.  Gerimedica can only
store one email address per user. Ideally one email address will be marked as
'primary' and if so, this will be saved. If no email address is marked as
primary and more than one email address is supplied, only one will be saved.
The email address saved will prioritize the 'work' over the 'home' and
'other' email addresses. If more than one of a given type is supplied, the
first email address with the highest priority based upon its type will be
```

```
accepted.",
        "required": false,
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "none",
        "subAttributes" : [
          {
            "name": "value",
            "type": "string",
            "multiValued": false,
            "description": "The email address value of the user. The maximum
size for this field is 255 characters",
            "required": false,
            "caseExact": false,
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "primary",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value indicating the 'primary'
or preferred email address of this user.  The primary attribute value 'true'
MUST appear no more than once.",
            "required": false,
            "mutability": "readWrite",
            "returned": "never"
          }
        ]
      },
      {
        "name": "phoneNumbers",
        "type": "complex",
        "multiValued": true,
        "description": "Phone numbers for the User. Gerimedica can save both
a mobile phone number and another phone number. Canonical type values of
'work' and 'mobile' will be accepted. If more than one value of a given type
is supplied, one should be marked with 'primary' and will be prioritized. If
more than one value of a given type is supplied without being marked as
'primary', the first will be accepted. If more than one phone number of a
given type is supplied and more than one is marked as 'primary', the first
supplied will be accepted.",
        "required": false,
        "mutability": "readWrite",
        "returned": "default",
```

```
        "subAttributes": [
          {
            "name": "value",
            "type": "string",
            "multiValued": false,
            "description": "Phone number of the User.",
            "required": false,
            "caseExact": false,
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "type",
            "type": "string",
            "multiValued": false,
            "description": "A label indicating the attribute's function,
e.g., 'work', or 'mobile'.",
            "required": false,
            "caseExact": false,
            "canonicalValues": [
              "work",
              "mobile"
            ],
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "primary",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value indicating the 'primary'
or preferred phone number. The primary attribute value 'true' MUST appear no
more than once per type.",
            "required": false,
            "mutability": "readWrite",
            "returned": "never"
          }
        ]
      },
      {
        "name": "gender",
        "type": "complex",
        "multiValued": false,
        "description":"An indication of the User’s gender. If no one selected
```

```
it will be given as UNKNOWN by default.",
        "required": false,
        "canonicalValues": [
          "UNKNOWN",
          "MALE",
          "FEMALE"
        ],
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "exportTimelineEvents",
        "type": "boolean",
        "multiValued": false,
        "description": "A Boolean value indicating whether to show the status
of the User's events on calendar or not.",
        "required": false,
        "caseExact": false,
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "roles",
        "type": "complex",
        "multiValued": true,
        "description": "A list of roles for the User that collectively
represent what activities the User will be undertaking while using this
system. A role is used to identify a set of authorizations the user is
granted within the system.",
        "required": false,
        "mutability": "readWrite",
        "returned": "default",
        "subAttributes": [
          {
            "name": "value",
            "type": "string",
            "multiValued": false,
            "description": "The unique identifier for the role",
            "required": false,
            "caseExact": true,
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          },
```

```
          {
           "name": "displayName",
           "type": "string",
           "multiValued": false,
           "description": "The name of the role.",
           "required": false,
           "caseExact": true,
           "mutability": "readWrite",
           "returned": "default",
           "uniqueness": "none"
          }
        ]
      },
      {
        "name": "entitlements",
        "type": "complex",
        "multiValued": true,
        "description": "A list of authorizations that the user will be able
to undertake within the system. Valid entitlement values must be one of the
canonical values.",
        "required": false,
        "mutability": "readWrite",
        "returned": "default",
        "subAttributes": [
          {
            "name": "value",
            "type": "string",
            "multiValued": false,
            "description": "The value of an entitlement.",
            "required": false,
            "caseExact": false,
            "canonicalValues: [
               "AccorderenGRZBasisGGZELV",
               "AccorderenWlz",
               "AccorderenWmo",
               "BehandelplanAccorderen",
               "BehandelrelatieWijzigen",
               "BeheerAdviessjablonen",
               "BeheerBriefsjablonen",
               "BeheerMedewerkers",
               "BeheerOverig",
               "BeheerTijdregistratie",
               "BeheerUitvalbak",
               "DBCZorgactiviteitenRegistreren",
               "DBCZorgtrajectenAanmakenBewerkenVerwijderen",
               "DeclarerenGRZBasiGGZELVZPM",
```

```
            "DeclarerenEerstelijnsParamedischeZorgHuisartsenzorGZSP",
            "DeclarerenWlz",
            "DeclarerenWmo",
            "DocumentenBeherenVoorAlleDisciplines",
            "DocumentenBeherenVoorEigenDiscipline",
            "DocumentenLezenVoorEigenDiscipline",
            "ELVTijdRegistreren",
            "HADeclaratie",
            "HAVerrichtingenWijzigen",
            "JuridischeInformatieBewerken",
            "LaboratoriumresultatenLezen",
            "MedischTabblad",
            "MeetinstrumentenInvullenBekijken",
            "MetingenVerwijderenNa24Uur",
            "MultidisciplinairTabblad",
            "PatiëntenMuteren",
            "RapportageGenereren",
            "RapportagesVerwijderenNa24Uur",
            "ToegangBonstat",
            "ToegangFarmedRx",
            "ToegangFierit",
            "ToegangHINQPortaal",
            "ToegangMedimo",
            "ToegangNCare",
            "ToegangQuestManagerROMPortaal",
            "ToegangRoosterplatform",
            "ToegangVIPLive",
            "ToegangViPharma",
            "ToegangZorgDomein",
            "ToevoegenBewerkenVerwijderenFinanciering",
            "ToevoegenBewerkenVerwijderenVerzekeringsgegevens",
            "VerslagenLezenActiviteitenbegeleiding",
            "VerslagenLezenDiëtetiek",
            "VerslagenLezenErgotherapie",
            "VerslagenLezenFysiotherapie",
            "VerslagenLezenGeestelijkeVerzorging",
            "VerslagenLezenLogopedie",
            "VerslagenLezenMaatschappelijkWerk",
            "VerslagenLezenMedischeDiscipline",
            "VerslagenLezenMondzorg",
            "VerslagenLezenMuziektherapie",
            "VerslagenLezenPsychologie",
            "VerslagenLezenPsychologieConsultatie",
            "VerslagenLezenPsychomotorischeTherapie",
            "VerslagenLezenVerpleegkunde",
            "VerslagenLezenZorg",
```

```
              "VerslagenToevoegen",
              "VertrouwelijkVerslagLezenActiviteitenbegeleiding",
              "VertrouwelijkVerslagLezenDiëtetiek",
              "VertrouwelijkVerslagLezenErgotherapie",
              "VertrouwelijkVerslagLezenFysiotherapie",
              "VertrouwelijkVerslagLezenGeestelijkeVerzorging",
              "VertrouwelijkVerslagLezenLogopedie",
              "VertrouwelijkVerslagLezenMaatschappelijkWerk",
              "VertrouwelijkVerslagLezenMedischeDiscipline",
              "VertrouwelijkVerslagLezenMondzorg",
              "VertrouwelijkVerslagLezenMuziektherapie",
              "VertrouwelijkVerslagLezenPsychologie",
              "VertrouwelijkVerslagLezenPsychologieConsultatie",
              "VertrouwelijkVerslagLezenPsychomotorischeTherapie",
              "VertrouwelijkVerslagLezenVerpleegkunde",
              "VerwerkenExterneVerwijzingen",
              "VerwijderenVragenlijsten",
              "VerwijzenNaarAndereAisciplines",
              "VolledigDossierPrinten",
              "ZorgplanAccorderen"
            ],
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          }
        ]
      }
    ],
    "meta" : {
      "resourceType": "Schema",
      "location":
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:User"
    }
  },
  {
    "id": "urn:ietf:params:scim:schemas:extension:ysis:2.0:User",
    "name": "User",
    "description": "Ysis extension for User ResourceType",
    "attributes": [
      {
        "name": "discipline",
        "type": "string",
        "multiValued": false,
        "description": "Identifier to indicate the Ysis Discipline of this
User.  Each User MUST include a non-empty discipline value, which must belong
to the set of canonical values.",
```

```
        "required": true,
        "caseExact": false,
        "canonicalValues": [
               "ACTIVITY_COACH",
               "ADMINISTRATIVE",
               "CARE",
               "DENTAL_CARE",
               "DIETITIAN",
               "ERGOTHERAPIST",
               "LOGOPEDIST",
               "MEDICAL",
               "MUSIC_THERAPIST",
               "NURSE",
               "PASTOR",
               "PHYSIOTHERAPIST",
               "PMT",
               "PSYCHOLOGIST",
               "PSYCHOLOGIST_CONSULTANT",
               "READ_ONLY",
               "SOCIAL_WORK"
        ],
        "mutability": "immutable",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "ysisInitials",
        "type": "string",
        "multiValued": false,
        "description": "Ysis Initials are used to uniquely identify a user
within the Ysis system. Each User MUST have a unique non-empty Ysis initials
value. The value should consist of alphanumeric characters. the maximum size
for this field is 10 characters",
        "required": true,
        "caseExact": false,
        "mutability": "immutable",
        "returned": "default",
        "uniqueness": "server"
      },
      {
        "name": "initials",
        "type": "string",
        "multiValued": false,
        "description": "Initials are used to identify a user within the Ysis
system. Each User MUST have a unique non-empty value. The value should
consist of alphanumeric characters. the maximum size for this field is 10
characters",
        "required": true,
```

```
        "caseExact": true,
        "mutability": "immutable",
        "returned": "default",
        "uniqueness": "none"
            },
      {
        "name": "bigNumber",
        "type": "string",
        "multiValued": false,
        "description": "Bignumber of the User. The maximum size for this
field is 11 characters.",
        "required": false,
        "caseExact": true,
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "server"
      },
      {
         "name": "position",
          "type": "string",
          "multiValued": false,
          "description": "The function of the user. The maximum size for this
field is 255 character.",
          "required": false,
          "caseExact": true,
          "mutability": "readWrite",
          "returned": "default",
          "uniqueness": "none"
      },
      {
         "name": "modules",
          "type": "string",
          "multiValued": true,
          "description": "Set of identifiers representing the YSIS Modules to be
enabled for this User. The set of modules can be empty.",
          "required": false,
          "caseExact": true,
          "canonicalValues": [
               "ECD_ADMIN",
               "ECD_CARE",
               "FINANCIAL",
               "FINANCIAL_EXPORT_PORTAL",
               "USER_MANAGEMENT",
               "YSIS_CORE",
               "YSIS_DBC"
          ],
```

```
          "mutability": "readWrite",
          "returned": "default",
          "uniqueness": "none"
      }
 ],
    "meta" : {
      "resourceType": "Schema",
      "location":
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:extension:ysis:2.0:User"
    }
  },
  {
    "id": "urn:ietf:params:scim:schemas:core:2.0:ResourceType",
    "name": "ResourceType",
    "description": "Describes a SCIM Resource Type.",
    "attributes": [
      {
        "name": "id",
        "type": "string",
        "multiValued": false,
        "description": "The resource type's server unique id.",
        "required": false,
        "caseExact": false,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "server"
      },
      {
        "name": "name",
        "type": "string",
        "multiValued": false,
        "description": "The resource type's name. e.g., 'user'.",
        "required": true,
        "caseExact": false,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "description",
        "type": "string",
        "multiValued": false,
        "description": "The resource type's human-readable description.",
        "required": false,
        "caseExact": false,
```

```
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "endpoint",
        "type": "reference",
        "referenceTypes": ["uri"],
        "multiValued": false,
        "description": "The resource type's HTTP-addressable endpoint
relative to the Base URL, e.g., '/users'.",
        "required": true,
        "caseExact": false,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "schema",
        "type": "reference",
        "referenceTypes" : ["uri"],
        "multiValued": false,
        "description": "The resource type's primary/base schema URI.",
        "required": true,
        "caseExact": true,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "schemaExtensions",
        "type": "complex",
        "multiValued": false,
        "description": "A list of URIs of the resource type's schema
extensions.",
        "required": true,
        "mutability": "readOnly",
        "returned": "default",
        "subAttributes" : [
          {
            "name": "schema",
            "type": "reference",
            "referenceTypes": ["uri"],
            "multiValued": false,
            "description": "The URI of a schema extension.",
            "required": true,
```

```
            "caseExact": true,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "required",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value that specifies whether
              or not the schema extension is required for the
              resource type.  If true, a resource of this type MUST
              include this schema extension and also include any
              attributes declared as required in this schema extension.
              If false, a resource of this type MAY omit this schema
              extension.",
            "required": true,
            "mutability": "readOnly",
            "returned": "default"
          }
        ]
      }
    ],
    "meta" : {
      "resourceType": "Schema",
      "location" :
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:ResourceType
"
    }
  },
  {
    "id": "urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig",
    "name": "ServiceProviderConfig",
    "description": "Schema representing the service provider's
configuration.",
    "attributes" : [
      {
        "name": "documentationUri",
        "type": "reference",
        "referenceTypes" : ["external"],
        "multiValued": false,
        "description": "An HTTP-addressable URL pointing to the
          service provider's human-consumable help documentation.",
        "required": false,
        "caseExact": false,
        "mutability": "readOnly",
```

```
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "patch",
        "type": "complex",
        "multiValued": false,
        "description": "A complex type that specifies PATCH configuration
options.",
        "required": true,
        "mutability": "readOnly",
        "returned": "default",
        "subAttributes" : [
          {
            "name": "supported",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value specifying whether or not
              the operation is supported.",
            "required": true,
            "mutability": "readOnly",
            "returned": "default"
          }
        ]
      },
      {
        "name": "bulk",
        "type": "complex",
        "multiValued": false,
        "description": "A complex type that specifies bulk configuration
options.",
        "required": true,
        "mutability": "readOnly",
        "returned": "default",
        "subAttributes" : [
          {
            "name": "supported",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value specifying whether or not
              the operation is supported.",
            "required": true,
            "mutability": "readOnly",
            "returned": "default"
          },
          {
```

```
        "name": "maxOperations",
        "type": "integer",
        "multiValued": false,
        "description": "An integer value specifying the maximum
          number of operations.",
        "required": true,
        "mutability": "readOnly",
        "uniqueness": "none",
        "returned": "default"
      },
      {
        "name": "maxPayloadSize",
        "type": "integer",
        "multiValued": false,
        "description": "An integer value specifying the maximum
          payload size in bytes.",
        "required": true,
        "mutability": "readOnly",
        "uniqueness": "none",
        "returned": "default",          }
    ]
  },
  {
    "name": "filter",
    "type": "complex",
    "multiValued": false,
    "description": "A complex type that specifies FILTER options.",
    "required": true,
    "mutability": "readOnly",
    "returned": "default",
    "subAttributes": [
      {
        "name": "supported",
        "type": "boolean",
        "multiValued": false,
        "description": "A Boolean value specifying whether or not
          the operation is supported.",
        "required": true,
        "mutability": "readOnly",
        "returned": "default"
      },
      {
        "name": "maxResults",
        "type": "integer",
        "multiValued": false,
        "description": "An integer value specifying the maximum
```

```
            number of resources returned in a response.",
          "required": true,
          "mutability": "readOnly",
          "uniqueness": "none",
          "returned": "default"
        }
      ]
    },
    {
      "name": "changePassword",
      "type": "complex",
      "multiValued": false,
      "description": "A complex type that specifies configuration
        options related to changing a password.",
      "required": true,
      "mutability": "readOnly",
      "returned": "default",
      "subAttributes": [
        {
          "name": "supported",
          "type": "boolean",
          "multiValued": false,
          "description": "A Boolean value specifying whether or not
            the operation is supported.",
          "required": true,
          "mutability": "readOnly",
          "returned": "default"
        }
      ]
    },
    {
      "name": "sort",
      "type": "complex",
      "multiValued": false,
      "description": "A complex type that specifies sort result options.",
      "required": true,
      "mutability": "readOnly",
      "returned": "default",
      "subAttributes": [
        {
          "name": "supported",
          "type": "boolean",
          "multiValued": false,
          "description": "A Boolean value specifying whether or not
            the operation is supported.",
          "required": true,
```

```
            "mutability": "readOnly",
            "returned": "default"
          }
        ]
      },
      {
        "name": "authenticationSchemes",
        "type": "complex",
        "multiValued": true,
        "description": "A complex type that specifies supported
          authentication scheme properties.",
        "required": true,
        "mutability": "readOnly",
        "returned": "default",
        "subAttributes": [
          {
            "name": "name",
            "type": "string",
            "multiValued": false,
            "description": "The common authentication scheme name, e.g., HTTP
Basic.",
            "required": true,
            "caseExact": false,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "description",
            "type": "string",
            "multiValued": false,
            "description": "A description of the authentication scheme.",
            "required": true,
            "caseExact": false,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "specUri",
            "type": "reference",
            "referenceTypes": ["external"],
            "multiValued": false,
            "description": "An HTTP-addressable URL pointing to the
              authentication scheme's specification.",
            "required": false,
```

```
            "caseExact": false,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "documentationUri",
            "type": "reference",
            "referenceTypes": ["external"],
            "multiValued": false,
            "description": "An HTTP-addressable URL pointing to the
              authentication scheme's usage documentation.",
            "required": false,
            "caseExact": false,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          }
        ]
      }
    ],
    "meta" : {
      "resourceType": "Schema",
      "location" :
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:ServiceProvi
derConfig"
    }
  },
  {
    "id": "urn:ietf:params:scim:schemas:core:2.0:Schema",
    "name": "Schema",
    "description": "Specifies the schema that describes a SCIM schema.",
    "attributes": [
      {
        "name": "id",
        "type": "string",
        "multiValued": false,
        "description": "The unique URI of the schema.",
        "required": true,
        "caseExact": true,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "name",
```

```
        "type": "string",
        "multiValued": false,
        "description": "The schema's human-readable name. e.g., 'user'.",
        "required": true,
        "caseExact": false,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "description",
        "type": "string",
        "multiValued": false,
        "description": "A human-readable description of the schema.",
        "required": false,
        "caseExact": false,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "attributes",
        "type": "complex",
        "multiValued": true,
        "description": "A complex attribute that includes the attributes of a
schema.",
        "required": true,
        "mutability": "readOnly",
        "returned": "default",
        "subAttributes": [
          {
            "name": "name",
            "type": "string",
            "multiValued": false,
            "description": "The attribute's name.",
            "required": true,
            "caseExact": true,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "type",
            "type": "string",
            "multiValued": false,
            "description": "The attribute's data type.
```

```
              Valid values include 'string', 'complex', 'boolean',
              'decimal', 'integer', 'dateTime', 'reference'.",
            "required": true,
            "caseExact": false,
            "canonicalValues": [
              "string",
              "complex",
              "boolean",
              "decimal",
              "integer",
              "dateTime",
              "reference"
            ],
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "multiValued",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value indicating an attribute's
plurality.",
            "required": true,
            "mutability": "readOnly",
            "returned": "default"
          },
          {
            "name": "description",
            "type": "string",
            "multiValued": false,
            "description": "A human-readable description of the attribute.",
            "required": false,
            "caseExact": true,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "required",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value indicating whether or not the
attribute is required.",
            "required": false,
            "mutability": "readOnly",
```

```
            "returned": "default"
          },
          {
            "name": "canonicalValues",
            "type": "string",
            "multiValued": true,
            "description": "A collection of canonical values.",
            "required": false,
            "caseExact": true,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "caseExact",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value indicating whether or
              not a string attribute is case sensitive.",
            "required": false,
            "mutability": "readOnly",
            "returned": "default"
          },
          {
            "name": "mutability",
            "type": "string",
            "multiValued": false,
            "description": "Indicates whether or not an attribute is
modifiable.",
            "required": false,
            "caseExact": true,
            "canonicalValues": [
              "readOnly",
              "readWrite",
              "immutable",
              "writeOnly"
            ],
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "returned",
            "type": "string",
            "multiValued": false,
            "description": "Indicates when an attribute is returned
```

```
        in a response (e.g., to a query).",
      "required": false,
      "caseExact": true,
      "canonicalValues": [
        "always",
        "default",
        "never",
        "request"
      ],
      "mutability": "readOnly",
      "returned": "default",
      "uniqueness": "none"
    },
    {
      "name": "uniqueness",
      "type": "string",
      "multiValued": false,
      "description": "Indicates how unique a value must be.",
      "required": false,
      "caseExact": true,
      "canonicalValues": [
        "server",
        "global"
        "none",
      ],
      "mutability": "readOnly",
      "returned": "default",
      "uniqueness": "none"
    },
    {
      "name": "referenceTypes",
      "type": "string",
      "multiValued": true,
      "description": "Used only with an attribute of type
        'reference'.  Specifies a SCIM resourceType that a
        reference attribute MAY refer to, e.g., 'User'.",
      "required": false,
      "caseExact": true,
      "mutability": "readOnly",
      "returned": "default",
      "uniqueness": "none"
    },
    {
      "name": "subAttributes",
      "type": "complex",
      "multiValued": true,
```

```
            "description": "Used to define the sub-attributes of a complex
attribute.",
            "required": false,
            "mutability": "readOnly",
            "returned": "default",
            "subAttributes" : [
              {
                "name": "name",
                "type": "string",
                "multiValued": false,
                "description": "The attribute's name.",
                "required": true,
                "caseExact": true,
                "mutability": "readOnly",
                "returned": "default",
                "uniqueness": "none"
              },
              {
                "name": "type",
                "type": "string",
                "multiValued": false,
                "description": "The attribute's data type.
                  Valid values include 'string', 'complex', 'boolean',
                   'decimal', 'integer', 'dateTime', 'reference'.",
                "required": true,
                "caseExact": false,
                "canonicalValues": [
                  "string",
                  "complex",
                  "boolean",
                  "decimal",
                  "integer",
                  "dateTime",
                  "reference"
                ],
                "mutability": "readOnly",
                "returned": "default",
                "uniqueness": "none"
              },
              {
                "name": "multiValued",
                "type": "boolean",
                "multiValued": false,
                "description": "A Boolean value indicating an attribute's
plurality.",
                "required": true,
```

```
                "mutability": "readOnly",
                "returned": "default"
              },
              {
                "name": "description",
                "type": "string",
                "multiValued": false,
                "description": "A human-readable description of the
attribute.",
                "required": false,
                "caseExact": true,
                "mutability": "readOnly",
                "returned": "default",
                "uniqueness": "none"
              },
              {
                "name": "required",
                "type": "boolean",
                "multiValued": false,
                "description": "A Boolean value indicating whether or
                  not the attribute is required.",
                "required": false,
                "mutability": "readOnly",
                "returned": "default"
              },
              {
                "name": "canonicalValues",
                "type": "string",
                "multiValued": true,
                "description": "A collection of canonical values.",
                "required": false,
                "caseExact": true,
                "mutability": "readOnly",
                "returned": "default",
                "uniqueness": "none"
              },
              {
                "name": "caseExact",
                "type": "boolean",
                "multiValued": false,
                "description": "A Boolean value indicating whether or
                  not a string attribute is case sensitive.",
                "required": false,
                "mutability": "readOnly",
                "returned": "default"
              },
```

```
        {
          "name": "mutability",
          "type": "string",
          "multiValued": false,
          "description": "Indicates whether or not an
            attribute is modifiable.",
          "required": false,
          "caseExact": true,
          "canonicalValues": [
            "readOnly",
            "readWrite",
            "immutable",
            "writeOnly"
          ],
          "mutability": "readOnly",
          "returned": "default",
          "uniqueness": "none"
        },
        {
          "name": "returned",
          "type": "string",
          "multiValued": false,
          "description": "Indicates when an attribute is
            returned in a response (e.g., to a query).",
          "required": false,
          "caseExact": true,
          "canonicalValues": [
            "always",
            "default",
            "never",
            "request"
          ],
          "mutability": "readOnly",
          "returned": "default",
          "uniqueness": "none"
        },
        {
          "name": "uniqueness",
          "type": "string",
          "multiValued": false,
          "description": "Indicates how unique a value must be.",
          "required": false,
          "caseExact": true,
          "canonicalValues": [
            "none",
            "server",
```

```
                  "global"
                ],
                "mutability": "readOnly",
                "returned": "default",
                "uniqueness": "none"
              },
              {
                "name": "referenceTypes",
                "type": "string",
                "multiValued": true,
                "description": "Used only with an attribute of type
                  'reference'.  Specifies a SCIM resourceType that a
                  reference attribute MAY refer to, e.g., 'User'.",
                "required": false,
                "caseExact": true,
                "mutability": "readOnly",
                "returned": "default",
                "uniqueness": "none"
              }
            ]
          }
        ]
      }
    ],
    "meta" : {
      "resourceType": "Schema",
      "location" :
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:Schema"
    }
  },
{
 "id": "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User",
 "name": "Enterprise",
 "description": "Enterprise for User ResourceType.",
 "attributes": [
   {
     "name": "employeeNumber",
     "type": "string",
     "multiValued": false,
     "description": "The employee number of the user. The maximum size for
this field is 20 character.",
     "required": false,
     "caseExact": true,
     "mutability": "readOnly",
     "returned": "default",
     "uniqueness": "server"
```

```
    }
  ],
  "meta": {
    "resourceType": "Schema",
    "location":
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:extension:enterprise:
2.0:User"
  }
},
  {
  "id": "urn:ietf:params:scim:api:messages:2.0:Error",
  "name": "Error",
  "description": "An error encountered during a SCIM operation.",
  "attributes": [
    {
      "name": "id",
      "type": "string",
      "multiValued": false,
      "description": "A unique identifier for the error.",
      "required": false,
      "caseExact": true,
      "mutability": "immutable",
      "returned": "never",
      "uniqueness": "none"
    },
    {
      "name": "status",
      "type": "integer",
      "multiValued": false,
      "description": "The HTTP status code for the error.",
      "required": true,
      "caseExact": true,
      "mutability": "immutable",
      "returned": "default",
      "uniqueness": "none"
    },
    {
      "name": "detail",
      "type": "string",
      "multiValued": false,
      "description": "A human-readable detailed error message.",
      "required": false,
      "caseExact": false,
      "mutability": "immutable",
      "returned": "default",
      "uniqueness": "none"
```

```
    },
    {
      "name": "scimType",
      "type": "string",
      "multiValued": false,
      "description": "The SCIM error code.",
      "required": false,
      "caseExact": true,
      "mutability": "immutable",
      "returned": "default",
      "uniqueness": "none"
    },
    ],
    "meta" : {
      "resourceType": "Schema",
      "location" :
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:api:messages:2.0:Error"
    }
 }
]
```

Example Error Responses

Error Scenario: A request to this endpoint which contains filtering options will result in an error.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "scimType": "invalidFilter",
  "detail": "Filtering is not supported for this operation."
}
```

Error Scenario: A request to this endpoint is made for an invalid organization.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "detail": "Operation is not permitted based on the supplied authorization."
}
```

Error Scenario: A request is made to this endpoint from an organization with multiple groups.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
```

```
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations with multiple groups."
}
```

Error Scenario: A request is made to this endpoint from an organization with no group.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations without groups."
}
```

Schema Definition

Endpoint: **/scim/v3/schemas/{schemaUri}**
HTTP Operation: **GET**

A HTTP GET to this endpoint can be used to return a single SCIM Schema supported by Gerimedica.

This endpoint is read only.

Filtering, sorting and pagination are not supported.

The following values are valid for the path parameter **schemaUri**:

Schema Name	**schemaUri parameter value**
User	urn:ietf:params:scim:schemas:core:2.0:User
Ysis User Extension	urn:ietf:params:scim:schemas:extension:ysis:2.0:User
User Enterprise	urn:ietf:params:scim:schemas:extension:enterprise:2.0:User
ResourceType	urn:ietf:params:scim:schemas:core:2.0:ResourceType
Service Provider Config	urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig
Schema	urn:ietf:params:scim:schemas:core:2.0:Schema
Error	urn:ietf:params:scim:api:messages:2.0:Error
Group	urn:ietf:params:scim:schemas:core:2.0:Group

Example Successful Response - User

Example response for request **/scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:User**

```
HTTP/1.1 200 OK
Content-Type: application/json
{
    "id": "urn:ietf:params:scim:schemas:core:2.0:User",
    "name": "User",
    "description": "User Account",
    "attributes": [
      {
        "name": "userName",
        "type": "string",
        "multiValued": false,
```

```
        "description": "Unique identifier for the User, typically
used by the user to directly authenticate to the service provider.
Each User MUST include a non-empty userName value.  This identifier
MUST be unique across the service provider's entire set of Users.
The maximum size for this field is 40 characters.",
        "required": true,
        "caseExact": false,
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "server"
      },
      {
        "name": "name",
        "type": "complex",
        "multiValued": false,
        "description": "The components of the user's real name.",
        "required": false,
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "none"
        "subAttributes": [
          {
            "name": "familyName",
            "type": "string",
            "multiValued": false,
            "description": "The family name of the User, or
last name in most Western languages (e.g., 'Jensen' given the full
name 'Ms. Barbara Jensen'). The maximum size for this field is 100
characters.",
            "required": true,
            "caseExact": true,
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "givenName",
            "type": "string",
            "multiValued": false,
            "description": "The given name of the User, or
first name in most Western languages (e.g., 'Barbara' given the
full name 'Ms. Barbara Jensen'). The maximum size for this field is 80
characters.",
            "required": true,
            "caseExact": true,
            "mutability": "readWrite",
```

```
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "infix",
            "type": "string",
            "multiValued": false,
            "description": "It is a Tussenvoegsel in Dutch. In a family name
affix positioned between a person's given name and the main part of their
family name. For Example for name: Emma van der Sluss, van der is infix.",
",
            "required": false,
            "caseExact": false,
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          }
        ]
      },
      {
        "name": "active",
        "type": "boolean",
        "multiValued": false,
        "description": "A Boolean value indicating the User's administrative
status.",
        "required": false,
        "mutability": "readWrite",
        "returned": "default"
      },
      {
        "name": "password",
        "type": "string",
        "multiValued": false,
        "description": "The User's cleartext password.  This
attribute is intended to be used as a means to specify an initial
password when creating a new User. A password is OPTIONAL when creating a
user and must not be present otherwise. If the password is not present at
creation, it will be auto generated. A password must be a minimum of 12
characters and a maximum of 64 characters.  All characters are valid except
'/' and '\'.
",
        "required": false,
        "caseExact": true,
        "mutability": "writeOnly",
        "returned": "never",
        "uniqueness": "none"
```

```
      },
      {
        "name": "bigNumber",
        "type": "string",
        "multiValued": false,
        "description": "Bignumber of the User. The maximum size for this
field is 11 characters.",
        "required": false,
        "caseExact": true,
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "server"
      },
      {
         "name": "position",
          "type": "string",
          "multiValued": false,
          "description": "The function of the user. The maximum size for this
field is 255 character.",
          "required": false,
          "caseExact": true,
          "mutability": "readWrite",
          "returned": "default",
          "uniqueness": "none"
      },
      {
        "name": "emails",
        "type": "complex",
        "multiValued": true,
        "description": "Email addresses for the user.  Gerimedica can only
store one email address per user. Ideally one email address will be marked as
'primary' and if so, this will be saved. If no email address is marked as
primary and more than one email address is supplied, only one will be saved.
The email address saved will prioritize the 'work' over the 'home' and
'other' email addresses. If more than one of a given type is supplied, the
first email address with the highest priority based upon its type will be
accepted.",
        "required": false,
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "none",
        "subAttributes" : [
          {
            "name": "value",
            "type": "string",
            "multiValued": false,
```

```
            "description": "The email address value of the user. The maximum
size for this field is 255 characters",
            "required": false,
            "caseExact": false,
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "primary",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value indicating the 'primary'
or preferred email address of this user.  The primary attribute value 'true'
MUST appear no more than once.",
            "required": false,
            "mutability": "readWrite",
            "returned": "never"
          }
        ]
      },
      {
        "name": "phoneNumbers",
        "type": "complex",
        "multiValued": true,
        "description": "Phone numbers for the User. Gerimedica can save both
a mobile phone number and another phone number. Canonical type values of
'work' and 'mobile' will be accepted. If more than one value of a given type
is supplied, one should be marked with 'primary' and will be prioritized. If
more than one value of a given type is supplied without being marked as
'primary', the first will be accepted. If more than one phone number of a
given type is supplied and more than one is marked as 'primary', the first
supplied will be accepted.",
        "required": false,
        "mutability": "readWrite",
        "returned": "default",
        "subAttributes": [
          {
            "name": "value",
            "type": "string",
            "multiValued": false,
            "description": "Phone number of the User.",
            "required": false,
            "caseExact": false,
            "mutability": "readWrite",
            "returned": "default",
```

```
            "uniqueness": "none"
          },
          {
            "name": "type",
            "type": "string",
            "multiValued": false,
            "description": "A label indicating the attribute's function,
e.g., 'work', or 'mobile'.",
            "required": false,
            "caseExact": false,
            "canonicalValues": [
              "work",
              "mobile"
            ],
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "primary",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value indicating the 'primary'
or preferred phone number. The primary attribute value 'true' MUST appear no
more than once per type.",
            "required": false,
            "mutability": "readWrite",
            "returned": "never"
          }
        ]
      },
      {
        "name": "roles",
        "type": "complex",
        "multiValued": true,
        "description": "A list of roles for the User that collectively
represent what activities the User will be undertaking while using this
system. A role is used to identify a set of authorizations the user is
granted within the system",
        "required": false,
        "mutability": "readWrite",
        "returned": "default",
        "subAttributes": [
          {
            "name": "value",
            "type": "string",
```

```
            "multiValued": false,
            "description": "The unique identifier for the role.",
            "required": false,
            "caseExact": true,
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          },
           {
            "name": "displayName",
            "type": "string",
            "multiValued": false,
            "description": "The name of the role.",
            "required": false,
            "caseExact": true,
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          }
        ]
      },
      {
        "name": "entitlements",
        "type": "complex",
        "multiValued": true,
        "description": "A list of authorizations that the user will be able
to undertake within the system. Valid entitlement values must be one of the
canonical values.",
        "required": false,
        "mutability": "readWrite",
        "returned": "default",
        "subAttributes": [
          {
            "name": "value",
            "type": "string",
            "multiValued": false,
            "description": "The value of an entitlement.",
            "required": false,
            "caseExact": false,
            "canonicalValues: [
               "AccorderenGRZBasisGGZELV",
               "AccorderenWlz",
               "AccorderenWmo",
               "BehandelplanAccorderen",
               "BehandelrelatieWijzigen",
               "BeheerAdviessjablonen",
```

```
                "BeheerBriefsjablonen",
                "BeheerMedewerkers",
                "BeheerOverig",
                "BeheerTijdregistratie",
                "BeheerUitvalbak",
                "DBCZorgactiviteitenRegistreren",
                "DBCZorgtrajectenAanmakenBewerkenVerwijderen",
                "DeclarerenGRZBasiGGZELVZPM",
                "DeclarerenEerstelijnsParamedischeZorgHuisartsenzorGZSP",
                "DeclarerenWlz",
                "DeclarerenWmo",
                "DocumentenBeherenVoorAlleDisciplines",
                "DocumentenBeherenVoorEigenDiscipline",
                "DocumentenLezenVoorEigenDiscipline",
                "ELVTijdRegistreren",
                "HADeclaratie",
                "HAVerrichtingenWijzigen",
                "JuridischeInformatieBewerken",
                "LaboratoriumresultatenLezen",
                "MedischTabblad",
                "MeetinstrumentenInvullenBekijken",
                "MetingenVerwijderenNa24Uur",
                "MultidisciplinairTabblad",
                "PatiëntenMuteren",
                "RapportageGenereren",
                "RapportagesVerwijderenNa24Uur",
                "ToegangBonstat",
                "ToegangFarmedRx",
                "ToegangFierit",
                "ToegangHINQPortaal",
                "ToegangMedimo",
                "ToegangNCare",
                "ToegangQuestManagerROMPortaal",
                "ToegangRoosterplatform",
                "ToegangVIPLive",
                "ToegangViPharma",
                "ToegangZorgDomein",
                "ToevoegenBewerkenVerwijderenFinanciering",
                "ToevoegenBewerkenVerwijderenVerzekeringsgegevens",
                "VerslagenLezenActiviteitenbegeleiding",
                "VerslagenLezenDiëtetiek",
                "VerslagenLezenErgotherapie",
                "VerslagenLezenFysiotherapie",
                "VerslagenLezenGeestelijkeVerzorging",
                "VerslagenLezenLogopedie",
                "VerslagenLezenMaatschappelijkWerk",
```

```
              "VerslagenLezenMedischeDiscipline",
              "VerslagenLezenMondzorg",
              "VerslagenLezenMuziektherapie",
              "VerslagenLezenPsychologie",
              "VerslagenLezenPsychologieConsultatie",
              "VerslagenLezenPsychomotorischeTherapie",
              "VerslagenLezenVerpleegkunde",
              "VerslagenLezenZorg",
              "VerslagenToevoegen",
              "VertrouwelijkVerslagLezenActiviteitenbegeleiding",
              "VertrouwelijkVerslagLezenDiëtetiek",
              "VertrouwelijkVerslagLezenErgotherapie",
              "VertrouwelijkVerslagLezenFysiotherapie",
              "VertrouwelijkVerslagLezenGeestelijkeVerzorging",
              "VertrouwelijkVerslagLezenLogopedie",
              "VertrouwelijkVerslagLezenMaatschappelijkWerk",
              "VertrouwelijkVerslagLezenMedischeDiscipline",
              "VertrouwelijkVerslagLezenMondzorg",
              "VertrouwelijkVerslagLezenMuziektherapie",
              "VertrouwelijkVerslagLezenPsychologie",
              "VertrouwelijkVerslagLezenPsychologieConsultatie",
              "VertrouwelijkVerslagLezenPsychomotorischeTherapie",
              "VertrouwelijkVerslagLezenVerpleegkunde",
              "VerwerkenExterneVerwijzingen",
              "VerwijderenVragenlijsten",
              "VerwijzenNaarAndereAisciplines",
              "VolledigDossierPrinten",
              "ZorgplanAccorderen"
            ],
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
          }
        ]
      }
    ],
    "meta" : {
      "resourceType": "Schema",
      "location" :
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:User"
    }
  }
```

In the v3 endpoints the User schema also contains the read-only attribute ‘groups’. See Scim Group Operations.

Example Successful Response - User Enterprise Extension

Example response for request:
scim/v3/schemas/urn:ietf:params:scim:schemas:extension:enterprise:2.0:User

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User",
  "name": "Enterprise",
  "description": "Enterprise for User ResourceType.",
  "attributes": [
  {
    "name": "employeeNumber",
    "type": "string",
    "multiValued": false,
    "description": "The employee number of the user. The maximum size for
this field is 20 character.",
    "required": false,
    "caseExact": true,
    "mutability": "readOnly",
    "returned": "default",
    "uniqueness": "server"
  }
  ],
  "meta": {
    "resourceType": "Schema",

"location":"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:extension:
enterprise:2.0:User"
  }
}
```

Example Successful Response - Ysis User Extension

Example response for request
/scim/v3/schemas/urn:ietf:params:scim:schemas:extension:ysis:2.0:User

```
HTTP/1.1 200 OK
Content-Type: application/json
{
    "id": "urn:ietf:params:scim:schemas:extension:ysis:2.0:User",
    "name": "User",
    "description": "Ysis extension for User ResourceType.",
    "attributes": [
      {
        "name": "discipline",
        "type": "string",
        "multiValued": false,
        "description": "Identifier to indicate the Ysis Discipline of this
User.  Each User MUST include a non-empty discipline value, which must belong
to the set of canonical values.",
        "required": true,
        "caseExact": true,
        "canonicalValues": [
               "ACTIVITY_COACH",
               "ADMINISTRATIVE",
               "CARE",
               "DENTAL_CARE",
               "DIETITIAN",
               "ERGOTHERAPIST",
               "LOGOPEDIST",
               "MEDICAL",
               "MUSIC_THERAPIST",
               "NURSE",
               "PASTOR",
               "PHYSIOTHERAPIST",
               "PMT",
               "PSYCHOLOGIST",
               "PSYCHOLOGIST_CONSULTANT",
               "READ_ONLY",
               "SOCIAL_WORK"
        ],
        "mutability": "immutable",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "ysisInitials",
        "type": "string",
        "multiValued": false,
```

```
        "description": "Ysis Initials are used to uniquely identify a user
within the Ysis system. Each User MUST have a unique non-empty Ysis initials
value. The value should consist of alphanumeric characters. the maximum size
for this field is 10 characters.",
        "required": true,
        "caseExact": false,
        "mutability": "immutable",
        "returned": "default",
        "uniqueness": "server"
            },
      {
        "name": "initials",
        "type": "string",
        "multiValued": false,
        "description": "Initials are used to identify a user within the Ysis
system. Each User MUST have a unique non-empty value. The value should
consist of alphanumeric characters. the maximum size for this field is 10
characters",
        "required": true,
        "caseExact": true,
        "mutability": "immutable",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "agbCode",
        "type": "string",
        "multiValued": false,
        "description": "AGB Code of this User. AGB Codes should be numeric
and 8 characters in length.",
        "required": false,
        "caseExact": true,
        "mutability": "readWrite",
        "returned": "default",
        "uniqueness": "none"
      },
      {
         "name": "modules",
          "type": "string",
          "multiValued": true,
          "description": "Set of identifiers representing the YSIS Modules to
enable for this User. The set of modules can be empty.",
          "required": false,
          "caseExact": true,
          "canonicalValues": [
               "ECD_ADMIN",
```

```
                "ECD_CARE",
                "FINANCIAL",
                "FINANCIAL_EXPORT_PORTAL",
                "USER_MANAGEMENT"
                "YSIS_CORE",
                "YSIS_DBC"
            ],
            "mutability": "readWrite",
            "returned": "default",
            "uniqueness": "none"
        }
    ],
    "meta" : {
      "resourceType": "Schema",
      "location":
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:extension:ysis:2.0:User"
    }
  }
```

Example Successful Response - Group

Example response for request /scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:Group

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "id": "urn:ietf:params:scim:schemas:core:2.0:Group",
  "name": "Group",
  "description": "A ward together with an access type: the users linked to the ward,
or the users authorized on it (the Group id encodes the access type, the location and
the ward). A linked user always has access to the ward; the authorized Group lists
only users given explicit access, so a user who is merely linked is not repeated in
it. Groups are read-only: they cannot be created or deleted (405) and their name
cannot be changed. Only the members can be changed, with PUT (replace all members) or
PATCH (add or remove members); a change is accepted and then applied asynchronously,
so a read right after it can still show the previous members for a few seconds. The
members are only returned when a single Group is retrieved. See the User resource's
read-only 'groups' attribute for the other direction.",
  "attributes": [
    {
      "name": "displayName",
      "type": "string",
      "multiValued": false,
      "description": "A human-readable name for the Group: the ward and location, and
whether it holds the linked or the authorized users. Groups are read-only: this value
cannot be changed (it is ignored on PUT, and rejected on PATCH).",
      "required": true,
```

```
      "caseExact": false,
      "mutability": "readOnly",
      "returned": "default",
      "uniqueness": "none"
    },
    {
      "name": "members",
      "type": "complex",
      "multiValued": true,
      "description": "The users in the Group. This is the only part of a Group that
can be changed: use PUT to replace all members, or PATCH to add or remove members. A
change is accepted and then applied asynchronously, so a read right after it can
still show the previous members for a few seconds. Included when a single Group is
retrieved (GET /Groups/{id}); the Group list leaves it out, as if
excludedAttributes=members had been sent. Gerimedica support accounts are not part of
this view and are never changed through it.",
      "required": false,
      "mutability": "readWrite",
      "returned": "default",
      "uniqueness": "none",
      "subAttributes": [
        {
          "name": "value",
          "type": "string",
          "multiValued": false,
          "description": "The id of the User that is a member of the Group.",
          "required": false,
          "caseExact": true,
          "mutability": "immutable",
          "returned": "default",
          "uniqueness": "none"
        },
        {
          "name": "$ref",
          "type": "reference",
          "multiValued": false,
          "description": "The URI of the User resource that is a member of the
Group.",
          "required": false,
          "caseExact": false,
          "mutability": "immutable",
          "returned": "default",
          "uniqueness": "none",
          "referenceTypes": ["User"]
        },
        {
          "name": "type",
          "type": "string",
          "multiValued": false,
          "description": "The type of the member. Only 'User' is supported: Groups
cannot be nested.",
```

```
          "required": false,
          "canonicalValues": ["User"],
          "caseExact": true,
          "mutability": "immutable",
          "returned": "default",
          "uniqueness": "none"
        },
        {
          "name": "display",
          "type": "string",
          "multiValued": false,
          "description": "A human-readable name for the member (the userName).
Ignored on write.",
          "required": false,
          "caseExact": false,
          "mutability": "readOnly",
          "returned": "default",
          "uniqueness": "none"
        }
      ]
    }
  ],
  "meta": {
    "resourceType": "Schema",
    "location":
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:Group"
  }
}
```

Example Successful Response - ResourceType

Example response for request
/scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:ResourceType

```
HTTP/1.1 200 OK
Content-Type: application/json
{
    "id": "urn:ietf:params:scim:schemas:core:2.0:ResourceType",
    "name": "ResourceType",
    "description": "Describes a SCIM Resource Type.",
    "attributes": [
      {
        "name": "id",
        "type": "string",
        "multiValued": false,
        "description": "The resource type's server unique id.",
        "required": false,
        "caseExact": false,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "server"
      },
      {
        "name": "name",
        "type": "string",
        "multiValued": false,
        "description": "The resource type's name. e.g., 'user'.",
        "required": true,
        "caseExact": false,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "description",
        "type": "string",
        "multiValued": false,
        "description": "The resource type's human-readable description.",
        "required": false,
        "caseExact": false,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "endpoint",
        "type": "reference",
```

```
        "referenceTypes": ["uri"],
        "multiValued": false,
        "description": "The resource type's HTTP-addressable endpoint
relative to the Base URL, e.g., '/users'.",
        "required": true,
        "caseExact": false,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "schema",
        "type": "reference",
        "referenceTypes" : ["uri"],
        "multiValued": false,
        "description": "The resource type's primary/base schema URI.",
        "required": true,
        "caseExact": true,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "schemaExtensions",
        "type": "complex",
        "multiValued": false,
        "description": "A list of URIs of the resource type's schema
extensions.",
        "required": true,
        "mutability": "readOnly",
        "returned": "default",
        "subAttributes" : [
          {
            "name": "schema",
            "type": "reference",
            "referenceTypes": ["uri"],
            "multiValued": false,
            "description": "The URI of a schema extension.",
            "required": true,
            "caseExact": true,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "required",
```

```
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value that specifies whether
              or not the schema extension is required for the
              resource type.  If true, a resource of this type MUST
              include this schema extension and also include any
              attributes declared as required in this schema extension.
              If false, a resource of this type MAY omit this schema
              extension.",
            "required": true,
            "mutability": "readOnly",
            "returned": "default"
          }
        ]
      }
    ],
    "meta" : {
      "resourceType": "Schema",
      "location" :
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:ResourceType
"
    }
  }
```

Example Successful Response - ServiceProviderConfig

Example response for request
/scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:ServerProviderConfig

```
HTTP/1.1 200 OK
Content-Type: application/json
{
    "id": "urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig",
    "name": "ServiceProviderConfig",
    "description": "Schema representing the service provider's
configuration.",
    "attributes" : [
      {
        "name": "documentationUri",
        "type": "reference",
        "referenceTypes" : ["external"],
        "multiValued": false,
        "description": "An HTTP-addressable URL pointing to the
          service provider's human-consumable help documentation.",
        "required": false,
        "caseExact": false,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "patch",
        "type": "complex",
        "multiValued": false,
        "description": "A complex type that specifies PATCH configuration
options.",
        "required": true,
        "mutability": "readOnly",
        "returned": "default",
        "subAttributes" : [
          {
            "name": "supported",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value specifying whether or not
              the operation is supported.",
            "required": true,
            "mutability": "readOnly",
            "returned": "default"
          }
        ]
      },
```

```
      {
        "name": "bulk",
        "type": "complex",
        "multiValued": false,
        "description": "A complex type that specifies bulk configuration
options.",
        "required": true,
        "mutability": "readOnly",
        "returned": "default",
        "subAttributes" : [
          {
            "name": "supported",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value specifying whether or not
              the operation is supported.",
            "required": true,
            "mutability": "readOnly",
            "returned": "default"
          },
          {
            "name": "maxOperations",
            "type": "integer",
            "multiValued": false,
            "description": "An integer value specifying the maximum
              number of operations.",
            "required": true,
            "mutability": "readOnly",
            "uniqueness": "none",
            "returned": "default"
          },
          {
            "name": "maxPayloadSize",
            "type": "integer",
            "multiValued": false,
            "description": "An integer value specifying the maximum
              payload size in bytes.",
            "required": true,
            "mutability": "readOnly",
            "uniqueness": "none",
            "returned": "default",          }
        ]
      },
      {
        "name": "filter",
        "type": "complex",
```

```
      "multiValued": false,
      "description": "A complex type that specifies FILTER options.",
      "required": true,
      "mutability": "readOnly",
      "returned": "default",
      "subAttributes": [
        {
          "name": "supported",
          "type": "boolean",
          "multiValued": false,
          "description": "A Boolean value specifying whether or not
            the operation is supported.",
          "required": true,
          "mutability": "readOnly",
          "returned": "default"
        },
        {
          "name": "maxResults",
          "type": "integer",
          "multiValued": false,
          "description": "An integer value specifying the maximum
            number of resources returned in a response.",
          "required": true,
          "mutability": "readOnly",
          "uniqueness": "none",
          "returned": "default"
        }
      ]
    },
    {
      "name": "changePassword",
      "type": "complex",
      "multiValued": false,
      "description": "A complex type that specifies configuration
        options related to changing a password.",
      "required": true,
      "mutability": "readOnly",
      "returned": "default",
      "subAttributes": [
        {
          "name": "supported",
          "type": "boolean",
          "multiValued": false,
          "description": "A Boolean value specifying whether or not
            the operation is supported.",
          "required": true,
```

```
            "mutability": "readOnly",
            "returned": "default"
          }
        ]
      },
      {
        "name": "sort",
        "type": "complex",
        "multiValued": false,
        "description": "A complex type that specifies sort result options.",
        "required": true,
        "mutability": "readOnly",
        "returned": "default",
        "subAttributes": [
          {
            "name": "supported",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value specifying whether or not
              the operation is supported.",
            "required": true,
            "mutability": "readOnly",
            "returned": "default"
          }
        ]
      },
      {
        "name": "authenticationSchemes",
        "type": "complex",
        "multiValued": true,
        "description": "A complex type that specifies supported
          authentication scheme properties.",
        "required": true,
        "mutability": "readOnly",
        "returned": "default",
        "subAttributes": [
          {
            "name": "name",
            "type": "string",
            "multiValued": false,
            "description": "The common authentication scheme name, e.g., HTTP
Basic.",
            "required": true,
            "caseExact": false,
            "mutability": "readOnly",
            "returned": "default",
```

```
            "uniqueness": "none"
          },
          {
            "name": "description",
            "type": "string",
            "multiValued": false,
            "description": "A description of the authentication scheme.",
            "required": true,
            "caseExact": false,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "specUri",
            "type": "reference",
            "referenceTypes": ["external"],
            "multiValued": false,
            "description": "An HTTP-addressable URL pointing to the
              authentication scheme's specification.",
            "required": false,
            "caseExact": false,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "documentationUri",
            "type": "reference",
            "referenceTypes": ["external"],
            "multiValued": false,
            "description": "An HTTP-addressable URL pointing to the
              authentication scheme's usage documentation.",
            "required": false,
            "caseExact": false,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          }
        ]
      }
    ],
    "meta" : {
      "resourceType": "Schema",
      "location" :
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:ServiceProvi
```

```
derConfig"
    }
  }
```

Example Successful Response - Schema

Example response for request
/scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:Schema

```
HTTP/1.1 200 OK
Content-Type: application/json
{
    "id": "urn:ietf:params:scim:schemas:core:2.0:Schema",
    "name": "Schema",
    "description": "Specifies the schema that describes a SCIM schema.",
    "attributes": [
      {
        "name": "id",
        "type": "string",
        "multiValued": false,
        "description": "The unique URI of the schema.",
        "required": true,
        "caseExact": true,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "name",
        "type": "string",
        "multiValued": false,
        "description": "The schema's human-readable name. e.g., 'user'.",
        "required": true,
        "caseExact": false,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "description",
        "type": "string",
        "multiValued": false,
        "description": "A human-readable description of the schema.",
        "required": false,
        "caseExact": false,
        "mutability": "readOnly",
        "returned": "default",
        "uniqueness": "none"
      },
      {
        "name": "attributes",
        "type": "complex",
```

```
        "multiValued": true,
        "description": "A complex attribute that includes the attributes of a
schema.",
        "required": true,
        "mutability": "readOnly",
        "returned": "default",
        "subAttributes": [
          {
            "name": "name",
            "type": "string",
            "multiValued": false,
            "description": "The attribute's name.",
            "required": true,
            "caseExact": true,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "type",
            "type": "string",
            "multiValued": false,
            "description": "The attribute's data type.
              Valid values include 'string', 'complex', 'boolean',
              'decimal', 'integer', 'dateTime', 'reference'.",
            "required": true,
            "caseExact": false,
            "canonicalValues": [
              "string",
              "complex",
              "boolean",
              "decimal",
              "integer",
              "dateTime",
              "reference"
            ],
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "multiValued",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value indicating an attribute's
plurality.",
```

```
            "required": true,
            "mutability": "readOnly",
            "returned": "default"
          },
          {
            "name": "description",
            "type": "string",
            "multiValued": false,
            "description": "A human-readable description of the attribute.",
            "required": false,
            "caseExact": true,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "required",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value indicating whether or not the
attribute is required.",
            "required": false,
            "mutability": "readOnly",
            "returned": "default"
          },
          {
            "name": "canonicalValues",
            "type": "string",
            "multiValued": true,
            "description": "A collection of canonical values.",
            "required": false,
            "caseExact": true,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "caseExact",
            "type": "boolean",
            "multiValued": false,
            "description": "A Boolean value indicating whether or
              not a string attribute is case sensitive.",
            "required": false,
            "mutability": "readOnly",
            "returned": "default"
          },
```

```
          {
            "name": "mutability",
            "type": "string",
            "multiValued": false,
            "description": "Indicates whether or not an attribute is
modifiable.",
            "required": false,
            "caseExact": true,
            "canonicalValues": [
              "readOnly",
              "readWrite",
              "immutable",
              "writeOnly"
            ],
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "returned",
            "type": "string",
            "multiValued": false,
            "description": "Indicates when an attribute is returned
              in a response (e.g., to a query).",
            "required": false,
            "caseExact": true,
            "canonicalValues": [
              "always",
              "default",
              "never",
              "request"
            ],
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "uniqueness",
            "type": "string",
            "multiValued": false,
            "description": "Indicates how unique a value must be.",
            "required": false,
            "caseExact": true,
            "canonicalValues": [
              "server",
              "global"
```

```
              "none",
            ],
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "referenceTypes",
            "type": "string",
            "multiValued": true,
            "description": "Used only with an attribute of type
              'reference'.  Specifies a SCIM resourceType that a
              reference attribute MAY refer to, e.g., 'User'.",
            "required": false,
            "caseExact": true,
            "mutability": "readOnly",
            "returned": "default",
            "uniqueness": "none"
          },
          {
            "name": "subAttributes",
            "type": "complex",
            "multiValued": true,
            "description": "Used to define the sub-attributes of a complex
attribute.",
            "required": false,
            "mutability": "readOnly",
            "returned": "default",
            "subAttributes" : [
              {
                "name": "name",
                "type": "string",
                "multiValued": false,
                "description": "The attribute's name.",
                "required": true,
                "caseExact": true,
                "mutability": "readOnly",
                "returned": "default",
                "uniqueness": "none"
              },
              {
                "name": "type",
                "type": "string",
                "multiValued": false,
                "description": "The attribute's data type.
                  Valid values include 'string', 'complex', 'boolean',
```

```
                  'decimal', 'integer', 'dateTime', 'reference'.",
                "required": true,
                "caseExact": false,
                "canonicalValues": [
                  "string",
                  "complex",
                  "boolean",
                  "decimal",
                  "integer",
                  "dateTime",
                  "reference"
                ],
                "mutability": "readOnly",
                "returned": "default",
                "uniqueness": "none"
              },
              {
                "name": "multiValued",
                "type": "boolean",
                "multiValued": false,
                "description": "A Boolean value indicating an attribute's
plurality.",
                "required": true,
                "mutability": "readOnly",
                "returned": "default"
              },
              {
                "name": "description",
                "type": "string",
                "multiValued": false,
                "description": "A human-readable description of the
attribute.",
                "required": false,
                "caseExact": true,
                "mutability": "readOnly",
                "returned": "default",
                "uniqueness": "none"
              },
              {
                "name": "required",
                "type": "boolean",
                "multiValued": false,
                "description": "A Boolean value indicating whether or
                  not the attribute is required.",
                "required": false,
                "mutability": "readOnly",
```

```
      "returned": "default"
    },
    {
      "name": "canonicalValues",
      "type": "string",
      "multiValued": true,
      "description": "A collection of canonical values.",
      "required": false,
      "caseExact": true,
      "mutability": "readOnly",
      "returned": "default",
      "uniqueness": "none"
    },
    {
      "name": "caseExact",
      "type": "boolean",
      "multiValued": false,
      "description": "A Boolean value indicating whether or
        not a string attribute is case sensitive.",
      "required": false,
      "mutability": "readOnly",
      "returned": "default"
    },
    {
      "name": "mutability",
      "type": "string",
      "multiValued": false,
      "description": "Indicates whether or not an
        attribute is modifiable.",
      "required": false,
      "caseExact": true,
      "canonicalValues": [
        "readOnly",
        "readWrite",
        "immutable",
        "writeOnly"
      ],
      "mutability": "readOnly",
      "returned": "default",
      "uniqueness": "none"
    },
    {
      "name": "returned",
      "type": "string",
      "multiValued": false,
      "description": "Indicates when an attribute is
```

```
            returned in a response (e.g., to a query).",
          "required": false,
          "caseExact": true,
          "canonicalValues": [
            "always",
            "default",
            "never",
            "request"
          ],
          "mutability": "readOnly",
          "returned": "default",
          "uniqueness": "none"
        },
        {
          "name": "uniqueness",
          "type": "string",
          "multiValued": false,
          "description": "Indicates how unique a value must be.",
          "required": false,
          "caseExact": true,
          "canonicalValues": [
            "none",
            "server",
            "global"
          ],
          "mutability": "readOnly",
          "returned": "default",
          "uniqueness": "none"
        },
        {
          "name": "referenceTypes",
          "type": "string",
          "multiValued": true,
          "description": "Used only with an attribute of type
            'reference'.  Specifies a SCIM resourceType that a
            reference attribute MAY refer to, e.g., 'User'.",
          "required": false,
          "caseExact": true,
          "mutability": "readOnly",
          "returned": "default",
          "uniqueness": "none"
        }
      ]
    }
  ]
}
```

```
    ],
    "meta" : {
      "resourceType": "Schema",
      "location" :
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:schemas:core:2.0:Schema"
    }
  }
```

Example Successful Response - Error

Example response for request **/scim/v3/schemas/urn:ietf:params:scim:api:messages:2.0:Error**

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "id": "urn:ietf:params:scim:api:messages:2.0:Error",
  "name": "Error",
  "description": "An error encountered during a SCIM operation.",
  "attributes": [
    {
      "name": "id",
      "type": "string",
      "multiValued": false,
      "description": "A unique identifier for the error.",
      "required": false,
      "caseExact": true,
      "mutability": "immutable",
      "returned": "never",
      "uniqueness": "none"
    },
    {
      "name": "status",
      "type": "integer",
      "multiValued": false,
      "description": "The HTTP status code for the error.",
      "required": true,
      "caseExact": true,
      "mutability": "immutable",
      "returned": "default",
      "uniqueness": "none"
    },
    {
      "name": "detail",
      "type": "string",
      "multiValued": false,
      "description": "A human-readable detailed error message.",
      "required": false,
      "caseExact": false,
      "mutability": "immutable",
      "returned": "default",
      "uniqueness": "none"
    },
    {
      "name": "scimType",
      "type": "string",
      "multiValued": false,
```

```
      "description": "The SCIM error code.",
      "required": false,
      "caseExact": true,
      "mutability": "immutable",
      "returned": "default",
      "uniqueness": "none"
    }
  ],
    "meta" : {
      "resourceType": "Schema",
      "location" :
"$BASE_URL/scim/v3/schemas/urn:ietf:params:scim:api:messages:2.0:Error"
    }
 }
```

Example Error Responses

Error Scenario: A request is sent to this endpoint containing filtering options.
Error Example:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "scimType": "invalidFilter",
  "detail": "Filtering is not supported for this operation."
}
```

Error Scenario: A request is sent to this endpoint with an invalid path parameter.
Error Example:

```
HTTP/1.1 400 BAD_REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidValue",
  "detail": "The path parameter 'urn:ietf:params:scim:schemas:core:2.0:NA' for
'schemaUri' is not supported."
}
```

Error Scenario: A request to this endpoint is made for an invalid organization.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
```

```
  "status": 403,
  "detail": "Operation is not permitted based on the supplied authorization."
}
```

Error Scenario: A request is made to this endpoint from an organization with multiple groups.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations with multiple groups."
}
```

Error Scenario: A request is made to this endpoint from an organization with no group.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations without groups."
}
```

Scim User Operations

Creating Users

Endpoint: **$BASE_URL/scim/v3/users**
HTTP Operation: **POST**
Authorization: **Bearer Token**

A HTTP POST to this endpoint can be used to create a Gerimedica User for the caller's organization. Error responses will be returned if invalid values are attempted to be set. See the 'Schema Definition - User' and 'Schema Definition - Error' sections for the full set of field requirements.

Filtering, sorting, and pagination are not supported.

Example Request

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User",
    "urn:ietf:params:scim:schemas:extension:ysis:2.0:User",
    "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
  ],
  "userName": "bjensen",
  "name": {
     "familyName": "Jensen",
     "givenName": "Barbara",
     "infix": "van"
  },
  "gender": null,
  "exportTimelineEvents": true,
  "bigNumber": null,
  "urn:ietf:params:scim:schemas:extension:ysis:2.0:User": {
     "ysisInitials": "Y.S",
     "discipline": "MEDICAL",
     "initials": "YS",
     "agbCode": "agbCode1",
     "modules": ["YSIS_CORE", "USER_MANAGEMENT"]
 },
 "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
     "employeeNumber": "2015010213058"
 },
 "emails": [
    {
      "value": "bjensen@example.com",
      "primary": true,
      "type": "home"
    }
```

```
  ],
  "phoneNumbers": [
    {
      "value": "555-555-5555",
      "type": "work"
    },
    {
      "value": "555-555-4444",
      "type": "mobile"
    }
  ],
  "password": "t1meMa$heen",
  "roles":[
     {
     "value": "51bc045c-d17c-4d29-bd3f-9b1avcf5a011",
     "displayName": "Fysiotherapie"
    },
    {
      "value": "13y79ab5-ec65-4733-95e5-49b5977733c7",
      "displayName": "Psychologie"
    }
  ],
  "entitlements":[
    {
      "value": "BehandelplanAccorderen"
    },
    {
      "value": "BehandelrelatieWijzigen"
    }
 ]
}
```

Example Successful Response

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
  "id": "2819c223-7f76-453a-919d-413861904646",
  "userName": "bjensen",
  "name": {
    "familyName": "Jensen",
    "givenName": "Barbara",
    "infix": "van"
  },
  "gender": "UNKNOWN",
  "exportTimelineEvents": true,
  "bigNumber": null,
  "groups": [],
  "urn:ietf:params:scim:schemas:extension:ysis:2.0:User": {
```

```
    "ysisInitials": "Y.S",
    "discipline": "MEDICAL",
    "agbCode": "agbCode1",
    "Initials": "YS",
    "modules": ["YSIS_CORE", "USER_MANAGEMENT"]
  },
  "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
    "employeeNumber": "2015010213058"
  },
  "emails": [
    {
      "value": "bjensen@example.com"
    }
  ],
  "phoneNumbers": [
    {
      "value": "555-555-5555",
      "type": "work"
    },
    {
      "value": "555-555-4444",
      "type": "mobile"
    }
  ],
  "active": true,
  "roles":[
    {
     "value": "51bc045c-d17c-4d29-bd3f-9b1avcf5a011",
     "displayName": "Fysiotherapie"
    },
    {
      "value": "13y79ab5-ec65-4733-95e5-49b5977733c7",
      "displayName": "Psychologie"
    }
  ],
  "entitlements":[
    {
      "value": "BehandelplanAccorderen"
    },
    {
      "value": "BehandelrelatieWijzigen"
    }
  ],
  "meta": {
    "resourceType": "User",
    "created": "2010-01-23T04:56:22Z",
    "lastModified": "2011-05-13T04:42:34Z",
    "location": "$BASE_URL/v3/users/2819c223-7f76-453a-919d-413861904646"
  }
}
```

Example Error Responses

Error Scenario: The caller attempts to create a User with a userName which already exists.
Example Error:

```
HTTP/1.1 409 Conflict
{
     "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
     "status": 409,
     "scimType": "uniqueness",
     "detail": "A User with the 'userName' 'bjensen' already exists."
}
```

Error Scenario: A request to this endpoint is made for an invalid organization.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "detail": "Operation is not permitted based on the supplied authorization."
}
```

Error Scenario: The caller attempts to create a User with a value which is invalid for a validated field.
Example Error:

```
HTTP/1.1 400 BAD_REQUEST
{
     "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
     "status": 400,
     "scimType": "invalidValue",
     "detail": "Invalid ysis initials, it should be max 10 character."
}
```

Error Scenario: A request is sent to this endpoint containing filtering options.
Error Example:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidFilter",
  "detail": "Filtering is not supported for this operation."
}
```

Error Scenario: A request is made to this endpoint from an organization with multiple groups.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations with multiple groups."
}
```

Error Scenario: A request is made to this endpoint from an organization with no group.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations without groups."
}
```

Re-activating Archived Users

A user which is deleted through the delete endpoint is archived, not removed. An archived user can be re-activated with the create endpoint.

To re-activate an archived user, send a create request (POST) with the userName, the ysisInitials and the discipline of the archived user. The user is then re-activated instead of created and keeps its userId. The response is the same as for a create (201 Created). The details of an archived user can be found with the list users endpoint and the parameter ‘includeArchived=true’, see Including Archived Users.

- The userName and the ysisInitials are compared without regard to case. Both must identify the same archived user and the discipline must match the discipline of the archived user.
- The user keeps its userId, ysisInitials, discipline, remarks and profession. All other attributes are taken from the request, as for a new user. A password in the request is ignored.
- If ‘active’ is false, the user is re-activated and blocked.
- The ward assignments of the user are removed when the user is archived. They are not restored. Assign the Groups again with the Group endpoints.
- A request which matches an archived user only partly, or which has another discipline, is refused with 409 Conflict. The detail explains what to change.

Error Scenario: The userName matches an archived user, but the ysisInitials do not.
Example Error:

```
HTTP/1.1 409 Conflict
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 409,
  "scimType": "uniqueness",
  "detail": "Cannot create user: userName 'bjensen' matches an archived user whose
```

```
ysisInitials is 'Y.S', but the request provides a different ysisInitials 'B.J'. To
re-activate this user, set ysisInitials to 'Y.S'. To create a new user instead, use a
userName and ysisInitials that are not already used by any user (active or archived)
in this organization."
}
```

Error Scenario: The userName and ysisInitials match an archived user, but the discipline does not.
Example Error:

```
HTTP/1.1 409 Conflict
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 409,
  "scimType": "uniqueness",
  "detail": "Cannot create user: userName 'bjensen' and ysisInitials 'Y.S' match an
archived user, but the requested discipline 'NURSE' does not match the archived
user's discipline 'MEDICAL'. To re-activate, use the matching discipline."
}
```

Retrieving Users (by userId)

Endpoint: **$BASE_URL/scim/v3/users/{user_id}**
HTTP Operation: **GET**
Authorization: **Bearer Token**

A HTTP GET call to this endpoint can be used to return an existing Gerimedica user in the SCIM JSON format. Only Users from the caller's organization can be returned. Attempting to return a user which does not exist, or which is archived, will return a 404 error. The query parameters 'attributes' and 'excludedAttributes' can be used to select the returned attributes, for example 'excludedAttributes=groups'.

The read-only attribute 'groups' lists the Groups of the user, see Groups of a User.

Filtering, sorting and pagination are not supported when returning a single user.

Example Request

```
HTTP/1.1 GET $BASE_URL/scim/v3/users/2819c223-7f76-453a-919d-413861904646
Authorization: Bearer xxx
Content-Type: application/scim+json
```

Example Successful Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Location: $BASE_URL/scim/v3/users/2819c223-7f76-453a-919d-413861904646
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User",
    "urn:ietf:params:scim:schemas:extension:ysis:2.0:User",
   "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
  ],
  "id": "2819c223-7f76-453a-919d-413861904646",
  "userName": "bjensen",
  "name": {
    "familyName": "Jensen",
    "givenName": "Barbara"
  },
  "emails": [
    {
      "value": "bjensen@example.com",
    }
  ],
  "phoneNumbers": [
    {
      "value": "555-555-5555",
      "type": "work"
    },
```

```
    {
      "value": "555-555-4444",
      "type": "mobile"
    }
  ],

  "active": true,
  "roles": [
     {
      "value": "51bc045c-d17c-4d29-bd3f-9b1avcf5a011",
      "displayName": "Fysiotherapie"
     },
     {
      "value": "13y79ab5-ec65-4733-95e5-49b5977733c7",
      "displayName": "Psychologie"
     }
  ],
  "entitlements": [
    {
      "value":  "BehandelplanAccorderen"
    },
    {
      "value":  "BehandelrelatieWijzigen"
    }
  ],
  "groups": [
    {
      "value": "linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12",
      "display": "Cardiology (Main building) - linked",
      "type": "direct",
      "$ref":
"$BASE_URL/scim/v3/Groups/linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12"
    }
  ],
"urn:ietf:params:scim:schemas:extension:ysis:2.0:User": {
   "ysisInitials": "Y.S",
    "discipline": "MEDICAL",
    "agbCode": "agbCode1",
    "initials": "YS"
 },
  "meta": {
    "resourceType": "User",
    "created": "2010-01-23T04:56:22Z",
    "lastModified": "2011-05-13T04:42:34Z",
    "location": "$BASE_URL/v3/users/2819c223-7f76-453a-919d-413861904646"
  }
}
```

Example Error Responses

Error Scenario: The caller attempts to get a user that does not exist.
Error Example:

```
HTTP/1.1 404 Not Found
{
     "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
     "status": 404,
     "detail":"Resource 2819c223-7f76-453a-919d-413861904646 not found"
}
```

Error Scenario: A request to this endpoint is made for an invalid organization.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "detail": "Operation is not permitted based on the supplied authorization."
}
```

Error Scenario: A request is sent to this endpoint containing filtering options.
Error Example:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidFilter",
  "detail": "Filtering is not supported for this operation."
}
```

Error Scenario: A request is made to this endpoint from an organization with multiple groups.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations with multiple groups."
}
```

Error Scenario: A request is made to this endpoint from an organization with no group.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
```

```
  "detail": "SCIM is not currently supported for organizations without groups."
}
```

Retrieving Users (with Filter)

Endpoint: **$BASE_URL/scim/v3/users**
HTTP Operation: **GET**
Authorization: **Bearer Token**

A HTTP GET call to this endpoint can be used to return a list of existing Gerimedica users in the SCIM JSON format. Only Users from the caller's organization can be returned. A filter can be provided to narrow the result, the currently supported filters can be found in the filtering section below. Archived users are not returned unless the parameter 'includeArchived=true' is passed, see Including Archived Users.

Sorting is not supported. Without a filter, the users are returned in pages of at most 100. Use the parameters 'startIndex' (starting at 1) and 'count' to select a page. With a filter, all matching users are returned unless 'count' is passed. The response contains the fields totalResults, startIndex and itemsPerPage. The query parameters 'attributes' and 'excludedAttributes' can be used to select the returned attributes. A 'startIndex' below 1 is treated as 1 and a negative 'count' as 0. With 'count=0' only 'totalResults' is returned.

Example Request

```
HTTP/1.1 GET
$BASE_URL/scim/v3/users?filter=urn:ietf:params:scim:schemas:extension:enterprise:2.0:
User:employeeNumber%20eq%20%221234%22
Authorization: Bearer xxx
Content-Type: application/scim+json
```

Example Successful Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Location: $BASE_URL/scim/v3/users/2819c223-7f76-453a-919d-413861904646
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
    "totalResults": 1,
  "Resources": [
   {
     "schemas": [
       "urn:ietf:params:scim:schemas:core:2.0:User",
       "urn:ietf:params:scim:schemas:extension:ysis:2.0:User",
      "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
     ],
     "id": "2819c223-7f76-453a-919d-413861904646",
     "userName": "bjensen",
     "name": {
       "familyName": "Jensen",
       "givenName": "Barbara"
     },
```

```
    "emails": [
      {
        "value": "bjensen@example.com",
      }
    ],
    "phoneNumbers": [
      {
        "value": "555-555-5555",
        "type": "work"
      },
      {
        "value": "555-555-4444",
        "type": "mobile"
      }
    ],
    "active": true,
    "roles": [
     {
       "value": "51bc045c-d17c-4d29-bd3f-9b1avcf5a011",
       "displayName": "Fysiotherapie"
     },
     {
       "value": "13y79ab5-ec65-4733-95e5-49b5977733c7",
       "displayName": "Psychologie"
     }
    ],
    "entitlements": [
      {
        "value":  "BehandelplanAccorderen"
      },
      {
        "value":  "BehandelrelatieWijzigen"
      }
    ],
   "urn:ietf:params:scim:schemas:extension:ysis:2.0:User": {
      "ysisInitials": "Y.S",
      "discipline": "MEDICAL",
      "agbCode": "agbCode1",
      "initials": "YS"
   },
  "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
     "employeeNumber": "1234"
  },
  "meta": {
     "resourceType": "User",
     "created": "2010-01-23T04:56:22Z",
     "lastModified": "2011-05-13T04:42:34Z",
     "location": "$BASE_URL/v3/users/2819c223-7f76-453a-919d-413861904646"
   }
  }
]
```

Example Error Responses

Error Scenario: The caller attempts to call with an incorrectly structured filter:
`$BASE_URL/scim/v3/users?filter=bad%20filter`
Error Example:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidFilter",
  "detail": "The filter 'bad filter' is not structured correctly."
}
```

Error Scenario: The caller attempts to with an invalid filter operator:
`$BASE_URL/scim/v3/users?filter=urn:ietf:params:scim:schemas:extension:enterprise:2.0:User:employeeNumber%20regex%20%221234%22`
Error Example:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidFilter",
  "detail": "The operator 'regex' is not supported."
}
```

Error Scenario: The caller attempts to with a filter value which is not enclosed in double quotes:
`$BASE_URL/scim/v3/users?filter=urn:ietf:params:scim:schemas:extension:enterprise:2.0:User:employeeNumber%20eq%201234`
Error Example:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidFilter",
  "detail": "The value to be filtered '1234' is not valid."
}
```

Error Scenario: A request is made to this endpoint from an organization with multiple groups.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations with multiple groups."
}
```

Error Scenario: A request is made to this endpoint from an organization with no group.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations without groups."
}
```

Including Archived Users

A user which is deleted is archived. Archived users are not returned by default. Add the query parameter ‘includeArchived=true’ to also return them. Archived users are returned after the active users and have ‘active’ set to false. The parameter can be used with all supported filters and without a filter.

Use it to find the userName, ysisInitials and discipline of an archived user, so that the user can be re-activated, see Creating Users.

Example Request

```
HTTP/1.1 GET
$BASE_URL/scim/v3/users?filter=userName%20eq%20%22bjensen%22&includeArchived=true
Authorization: Bearer xxx
Content-Type: application/scim+json
```

Example Successful Response

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 1,
  "startIndex": 1,
  "itemsPerPage": 1,
  "Resources": [
    {
      "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:User",
        "urn:ietf:params:scim:schemas:extension:ysis:2.0:User"
      ],
      "id": "2819c223-7f76-453a-919d-413861904646",
      "userName": "bjensen",
      "name": {
        "familyName": "Jensen",
```

```
        "givenName": "Barbara"
      },
      "active": false,
      "urn:ietf:params:scim:schemas:extension:ysis:2.0:User": {
        "ysisInitials": "Y.S",
        "discipline": "MEDICAL",
        "agbCode": "agbCode1",
        "initials": "YS"
      }
    }
  ]
}
```

Filtering

The current SCIM implementation supports a limited subset of filtering functionality. Filtering allows a user to pass in a filter as a query parameter to a supported endpoint. Any SCIM filtering option not specifically mentioned in this section is not currently supported.

Supported Filter Structure:

The SCIM implementation supports the following filter structure:

SchemaId:attributeName operator “value”

The SchemaId can be left out, for example: userName eq “bjensen”.

Supported SchemaIds and AttributeName:

We currently only support filtering on a single attribute at a time. The specified attribute must be supported and must belong to the specified schema. The current set of supported attributes and their associated schemas are:

Schema Id	Attribute Name
urn:ietf:params:scim:schemas:extension:enterprise:2.0:User	employeeNumber
urn:ietf:params:scim:schemas:extension:enterprise:2.0:User	userName

Supported Operators:

Operators are case insensitive. All other operators such as “regex”, “co”, “(“, “)” etc. are not supported and will cause undesired results or errors if attempted.

The current set of supported operators for this endpoint are:

Operator	Description
eq	Equals, exact match. When used in a filter, the ‘eq’ operator will match any objects which exactly match the filter value by querying the filter attribute within the specified filter schema. If the attribute contains multiple values, if any of the values match, the result will be returned.
pr	Present (has value). When used in a filter, the ‘pr’ operator will match any object which has a value for the specified attribute within the specified filter schema.

Supported Values:

The value must be enclosed in double quotes, an error will be returned if double quotes are missing. An empty value, eg “”, is not valid and returns an error.

This only applies to operators that accept a value.

Updating Users (PUT)

Endpoint: **$BASE_URL/scim/v3/users/{user_id}**
HTTP Operation: **PUT**
Authorization: **Bearer Token**

A HTTP PUT call to this endpoint can be used to update mutable attributes for a User. The intent of a PUT call is to replace the entire object rather than update individual fields. PUT calls cannot be used to create new resources. Only Users from the caller's organization can be updated.

If an attribute is immutable, the PUT call must contain the same value (or no value) as the existing object. A 400 bad request will be returned if modification of an immutable attribute is attempted.

If an attribute is mutable, supplying new value(s) will replace the old values. Supplying a null or empty value will remove the existing value if the attribute is optional and will result in an error if the attribute is mandatory. If no value currently exists for a mutable attribute, the new value(s) will be applied.

"writeOnly" attributes such as the User password cannot currently be modified via a PUT call. They should not be supplied and a 400 bad request error will be returned.

The read-only attribute 'groups' is ignored on a PUT. Change the Group memberships of a user with the Group endpoints. An archived user cannot be updated and returns a 404 error.

Filtering, sorting, and pagination are not supported.

Example Request

```
HTTP/1.1 PUT $BASE_URL/scim/v3/users/2819c223-7f76-453a-919d-413861904646
Authorization: Bearer xxx
Content-Type: application/scim+json
{
     "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
     "userName": "bjensen",
     "name":{
       "familyName": "Jensen-Smith",
       "givenName": "Barbara",
       "infix": "van"
     },
     "gender": "UNKNOWN",
     "exportTimelineEvents": true,
     "roles":[],
     "emails":[
       {
           "value": "bjensen@example.com"
       },
       {
           "value": "babs@jensen.org"
```

```
      }
    ]
}
```

Example Successful Response

```
{
    "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
    "id": "2819c223-7f76-453a-919d-413861904646",
    "userName": "bjensen",
    "name":{
      "familyName": "Jensen-Smith",
      "givenName": "Barbara",
      "infix": "van"
    },
    "gender": "UNKNOWN",
    "exportTimelineEvents": true,
    "emails": [
      {
          "value": "bjensen@example.com"
      },
      {
          "value": "babs@jensen.org"
      }
    ],
    "modules": [],
    "meta": {
      "resourceType": "User",
      "created": "2011-08-08T04:56:22Z",
      "lastModified": "2011-08-08T08:00:12Z",
      "location": "$BASE_URL/scim/v3/users/2819c223-7f76-453a-919d-413861904646"
    }
}
```

Example Error Response

Error Scenario: Attempting to update an attribute which must be unique to a value which already exists.
Example Error:

```
HTTP/1.1 409 Conflict
{
    "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
    "detail": "A User with the 'userName' 'bjensen' already exists.",
    "scimType": "uniqueness",
    "status": 409
}
```

Error Scenario: Attempting to pass in a JSON object which is not parseable.
Example Error:

```
HTTP/1.1 400 Bad Request
{
       "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
       "detail": "Request is unparsable, syntactically incorrect, or violates
schema.",
       "scimType": "invalidSyntax",
       "status": 400
 }
```

Error Scenario: Caller attempts to clear the value of a mandatory attribute.
Example Error:

```
HTTP/1.1 400 Bad Request
{
     "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
     "detail": "A value for the 'userName' attribute is required.",
     "scimType": "mutability",
     "status": 400
}
```

Error Scenario: Caller attempts to update an attribute which is readOnly
Example Error:

```
HTTP/1.1 400 Bad Request
{
     "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
     "detail": "The 'id' attribute is readOnly.",
     "scimType": "mutability",
     "status": 400
}
```

Error Scenario: Caller attempts to update an attribute which is immutable.
Example Error:

```
HTTP/1.1 400 Bad Request
{
     "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
     "detail": "The 'discipline' attribute is immutable.",
     "scimType": "mutability",
     "status": 400
}
```

Error Scenario: Caller attempts to update an attribute to an invalid value.
Example Error:

```
HTTP/1.1 400 Bad Request
{
     "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
     "detail": "The value 'DETECTIVE' is not valid for the 'roles' attribute.",
     "scimType": "invalidValue",
     "status": 400
w}
```

Error Scenario: Caller attempting to update a User which does not exist.
Example Error:

```
HTTP/1.1 404 Not Found
{
     "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
     "detail": "Resource '2819c223-7f76-453a-919d-413861904646' not found.",
     "status": 404
}
```

Error Scenario: A request is sent to this endpoint containing filtering options.
Error Example:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidFilter",
  "detail": "Filtering is not supported for this operation."
}
```

Error Scenario: A request to this endpoint is made for an invalid organization.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "detail": "Operation is not permitted based on the supplied authorization."
}
```

Error Scenario: A request is made to this endpoint from an organization with multiple groups.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations with multiple groups."
}
```

Error Scenario: A request is made to this endpoint from an organization with no group.

Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations without groups."
}
```

Updating Users (PATCH)

Endpoint: **$BASE_URL/scim/v3/users/{user_id}**
HTTP Operation: **PATCH**
Authorization: **Bearer Token**

A HTTP PATCH call to this endpoint can be used to change individual attributes of a User. Only Users from the caller's organization can be changed. The result is validated like a PUT, so the rules for immutable, readOnly and mandatory attributes of a PUT apply. If one operation fails, the User is not changed.

- The operations 'add', 'remove' and 'replace' are supported. The operation name is not case sensitive. The operations are applied in order.
- The path is an attribute ('userName'), a sub-attribute ('name.familyName'), an attribute of an extension with the schema id in front ('urn:ietf:params:scim:schemas:extension:enterprise:2.0:User:employeeNumber'), or a value of a multi-valued attribute selected with a filter ('emails[type eq "work"].value'). Only the operator 'eq' is supported in a path filter.
- The path can be left out when the value is an object with the attributes to change. A remove always needs a path.
- A remove on a multi-valued attribute with a list of values removes those values. A remove without a value removes the attribute. A mandatory attribute cannot be removed.
- Setting 'active' to false blocks the User, setting it to true activates the User again. The value can be a boolean or the string 'true' or 'false' in any case.
- Attributes which are not known to Gerimedica, and 'externalId', are ignored. The attributes 'id', 'meta', 'schemas' and 'groups' are readOnly: the same value is accepted, a different value is refused.
- The 'schemas' of the request can be left out.
- The response is the complete User, with status 200. The query parameters 'attributes' and 'excludedAttributes' can be used to select the returned attributes.

Filtering, sorting and pagination are not supported.

Example Request

```
HTTP/1.1 PATCH $BASE_URL/scim/v3/users/2819c223-7f76-453a-919d-413861904646
Authorization: Bearer xxx
Content-Type: application/scim+json
{
```

```
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "Operations": [
    {
      "op": "replace",
      "path": "active",
      "value": false
    },
    {
      "op": "replace",
      "path": "name.familyName",
      "value": "Jensen-Smith"
    },
    {
      "op": "add",
      "path": "phoneNumbers",
      "value": [
        {
          "value": "555-555-4444",
          "type": "mobile"
        }
      ]
    }
  ]
}
```

Example Successful Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Location: $BASE_URL/scim/v3/users/2819c223-7f76-453a-919d-413861904646
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User",
    "urn:ietf:params:scim:schemas:extension:ysis:2.0:User"
  ],
  "id": "2819c223-7f76-453a-919d-413861904646",
  "userName": "bjensen",
  "name": {
    "familyName": "Jensen-Smith",
    "givenName": "Barbara"
  },
  "phoneNumbers": [
    {
      "value": "555-555-5555",
      "type": "work"
    },
    {
      "value": "555-555-4444",
      "type": "mobile"
```

```
    }
  ],
  "active": false,
  "groups": [],
  "urn:ietf:params:scim:schemas:extension:ysis:2.0:User": {
    "ysisInitials": "Y.S",
    "discipline": "MEDICAL",
    "agbCode": "agbCode1",
    "initials": "YS"
  },
  "meta": {
    "resourceType": "User",
    "location": "$BASE_URL/scim/v3/users/2819c223-7f76-453a-919d-413861904646"
  }
}
```

Example Error Responses

The errors of a PUT, for example for an immutable attribute, an invalid value or a userName which already exists, also apply to a PATCH, see Updating Users (PUT).

Error Scenario: The caller attempts to change a readOnly attribute, for example ‘groups’ or ‘id’.
Example Error:

```
HTTP/1.1 400 Bad Request
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "mutability",
  "detail": "The 'groups' attribute is readOnly."
}
```

Error Scenario: The caller attempts to remove a mandatory attribute.
Example Error:

```
HTTP/1.1 400 Bad Request
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "mutability",
  "detail": "The 'userName' attribute is required and cannot be removed."
}
```

Error Scenario: The caller sends a remove operation without a path.
Example Error:

```
HTTP/1.1 400 Bad Request
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
```

```
  "scimType": "noTarget",
  "detail": "A remove operation needs a 'path'."
}
```

Error Scenario: The caller replaces a value through a filter which matches nothing.
Example Error:

```
HTTP/1.1 400 Bad Request
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "noTarget",
  "detail": "No value of 'emails' has type 'work'."
}
```

Error Scenario: The caller uses a filter operator other than ‘eq’ in a path.
Example Error:

```
HTTP/1.1 400 Bad Request
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidFilter",
  "detail": "The filter operator 'co' is not supported. Only 'eq' is."
}
```

Error Scenario: The caller sends a request without operations.
Example Error:

```
HTTP/1.1 400 Bad Request
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidSyntax",
  "detail": "A PATCH request needs at least one entry in 'Operations'."
}
```

Error Scenario: The caller attempts to change a User which does not exist, or which is archived.
Example Error:

```
HTTP/1.1 404 Not Found
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 404,
  "detail": "Resource '2819c223-7f76-453a-919d-413861904646' not found."
}
```

Error Scenario: A request is sent to this endpoint containing filtering options.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidFilter",
  "detail": "Filtering is not supported for this operation."
}
```

Error Scenario: A request to this endpoint is made for an invalid organization.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "detail": "Operation is not permitted based on the supplied authorization."
}
```

Error Scenario: A request is made to this endpoint from an organization with multiple groups.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations with multiple groups."
}
```

Error Scenario: A request is made to this endpoint from an organization with no group.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations without groups."
}
```

Deleting Users

Endpoint: **$BASE_URL/scim/v3/users/{user_id}**
HTTP Operation: **DELETE**
Authorization: **Bearer Token**

A HTTP DELETE call to this endpoint can be used to delete a Gerimedica User. Only Users from the caller's organization can be deleted. Attempting to delete a User which does not exist for the caller's organization will result in a 404 not found. Deleting a user archives the user. An archived user is removed from all wards (Group memberships), is no longer returned unless 'includeArchived=true' is passed to the list users endpoint, and can be re-activated, see Creating Users. The removal from the wards is processed asynchronously, see Asynchronous Processing.

Filtering, sorting and pagination are not supported.

Example Request

```
HTTP/1.1 DELETE $BASE_URL/scim/v3/users/2819c223-7f76-453a-919d-413861904646
Authorization: Bearer xxx
```

Example Successful Response

```
HTTP/1.1 204 No Content
```

Example Error Responses

Error Scenario: Attempting to delete a User which does not exist.
Example Error:

```
HTTP/1.1 404 Not Found
{
     "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
     "detail": "Resource '2819c223-7f76-453a-919d-413861904646' not found.",
     "status": 404
}
```

Error Scenario: A request is sent to this endpoint containing filtering options.
Error Example:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidFilter",
  "detail": "Filtering is not supported for this operation."
}
```

Error Scenario: A request to this endpoint is made for an invalid organization.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "detail": "Operation is not permitted based on the supplied authorization."
}
```

Error Scenario: A request is made to this endpoint from an organization with multiple groups.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations with multiple groups."
}
```

Error Scenario: A request is made to this endpoint from an organization with no group.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations without groups."
}
```

Scim Role Operations

Retrieving Roles

Endpoint: **$BASE_URL/scim/v3/roles**
HTTP Operation: **GET**
Authorization: **Bearer Token**

An HTTP GET to this endpoint can be used to retrieve roles.

Example Request

```
HTTP/1.1 GET $BASE_URL/scim/v3/roles
Authorization: Bearer xxx
Content-Type: application/scim+json
```

Example Successful Response

```
  "schemas": [
      "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 1,
  "Resources": [
    {
      "value": "39c0be3a-891a-4c77-a65e-49ee16f3538d",
      "displayName": "Diëtetiek"
    },
    {
      "value": "b3497775-9482-4e77-a240-8607db84fee5",
      "displayName": "Ergotherapie"
    },
    {
      "value": "ceb5c7bf-2092-4377-975a-7303af8e42a8",
      "displayName": "Geestelijke verzorging"
    },
    {
      "value": "135f9ab7-ec65-4777-95e5-49b5977c33c7",
      "displayName": "Muziektherapie"
    },
    {
      "value": "9c4ca071-eb2c-4767-a228-0150c727e3f4",
      "displayName": "Custom Role"
    },
  ]
```

Example Error Responses

Error Scenario: A request is made to this endpoint which contains filtering options will result in an error.

Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "scimType": "invalidFilter",
  "detail": "Filtering is not supported for this operation."
}
```

Error Scenario: A request is made to this endpoint from an organization with multiple groups.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations with multiple groups."
}
```

Error Scenario: A request is made to this endpoint from an organization with no group.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "detail": "SCIM is not currently supported for organizations without groups."
}
```

Error Scenario: A request to this endpoint is made for an invalid organization.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "detail": "Operation is not permitted based on the supplied authorization."
}
```

Scim Group Operations

Locations and Wards as Groups

Locations and wards are available as SCIM Groups. Every ward is represented by two Groups: one for the users linked to the ward and one for the users authorized for the ward. Assigning a user to a Group is the same flow as linking or authorizing the user manually in Ysis, and the same rules apply.

- Group id: ‘linked~{locationId}~{wardId}’ or ‘authorized~{locationId}~{wardId}’.
- Group displayName: the ward, the location and the type, for example ‘Cardiology (Main building) - linked’.
- Linked: the user works on the ward.
- Authorized: the user has access to the ward. A linked user always has access to the ward, so linked users are not repeated in the authorized Group.

Groups come from Ysis. They cannot be created, renamed or deleted through SCIM. Only the members of a Group can be changed, with PATCH (add or remove members) or PUT (replace all members). Members are users, identified by the userId which is returned when the user is created.

Use GET /Groups to get the location and ward information. It returns every ward of the organization, so the Groups can be mapped to the wards of your system before users are assigned.

Rules:

- The user must exist and must not be archived before it can be added to a Group.
- A user with the discipline ADMINISTRATIVE cannot be assigned to a ward.
- Adding a user to an authorized Group turns on the setting ‘Autorisaties inperken per afdeling’ for that user, as in the manual flow. The user then has access only to the linked and the authorized wards.
- The rules of Ysis for wards apply as in the manual flow. A change which Ysis refuses, for example removing the only doctor of a ward which has patients, is accepted by SCIM but not applied.

Assigning Users to Wards

A location has wards, and every ward is a Group. To assign a user to a ward, add the user to the Group of that ward. There is no Group for a location by itself: the Group of a ward already contains its location.

Step 1: Create the user
Create the user, or find the existing user, and keep the user id. See Creating Users and Retrieving Users (with Filter).

Step 2: Find the ward

Call GET $BASE_URL/scim/v3/Groups. Every ward is listed twice, once for linked users and once for authorized users, for example ‘Cardiology (Main building) - linked’ and ‘Cardiology (Main building) - authorized’. Choose the Group of the ward and the access type you need, and keep its id. Do this before every change, so that the ids and the members are up to date. See Locations and Wards as Groups for the difference between linked and authorized.

Step 3: Add the user to the Group
Send a PATCH to $BASE_URL/scim/v3/Groups/{group id} with the user id as a member. Several users can be added in one request. The Group id is in the URL and the user is in the request.

```
HTTP/1.1 PATCH
$BASE_URL/scim/v3/Groups/linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12
Authorization: Bearer xxx
Content-Type: application/scim+json
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "Operations": [
    {
      "op": "add",
      "path": "members",
      "value": [
        {
          "value": "2819c223-7f76-453a-919d-413861904646"
        }
      ]
    }
  ]
}
```

Step 4: Check the result
A 200 response means that the request is accepted, not that it is applied yet. A few seconds later, retrieve the Group, or retrieve the user and read the attribute ‘groups’, to confirm the change. See Asynchronous Processing.

Removing and moving users
To remove a user from a ward, send a PATCH with the operation ‘remove’, see Updating Group Members (PATCH). To move a user to another ward, remove the user from the old Group and add the user to the new Group. These are two requests, one for each Group.

Asynchronous Processing

Changes to the members of a Group are processed asynchronously. A successful response to a PATCH or a PUT means that the request is accepted. It does not mean that the change is applied. The change is applied a few seconds later.

A GET directly after a change can still show the previous members. Retrieve the Group again after a few seconds to confirm the change. Archiving a user removes the ward assignments of the user in the same way.

Always fetch the current location and ward information before sending a PATCH, because wards and their members can change in Ysis at any time.

Retrieving Groups (List)

Endpoint: **$BASE_URL/scim/v3/Groups**
HTTP Operation: **GET**
Authorization: **Bearer Token**

A HTTP GET call to this endpoint can be used to return the Groups (wards) of the caller's organization. Every ward has a linked and an authorized Group. The Groups are returned without members, retrieving a single Group to get its members.

Filtering, sorting and pagination are not supported.

Example Request

```
HTTP/1.1 GET $BASE_URL/scim/v3/Groups
Authorization: Bearer xxx
Content-Type: application/scim+json
```

Example Successful Response

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 2,
  "Resources": [
    {
      "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Group"],
      "id": "linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12",
      "displayName": "Cardiology (Main building) - linked",
      "meta": {
        "resourceType": "Group",
        "location":
"$BASE_URL/scim/v3/Groups/linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12"
      }
    },
    {
      "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Group"],
      "id": "authorized~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12",
      "displayName": "Cardiology (Main building) - authorized",
      "meta": {
```

```
        "resourceType": "Group",
        "location":
"$BASE_URL/scim/v3/Groups/authorized~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12"
      }
    }
  ]
}
```

Example Error Responses

Error Scenario: A request to this endpoint contains filtering options.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidFilter",
  "detail": "Filtering is not supported for this operation."
}
```

Error Scenario: A request to this endpoint is made for an invalid organization.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "detail": "Operation is not permitted based on the supplied authorization."
}
```

Error Scenario: The location and ward data is temporarily unavailable. Nothing was changed. Retry after the number of seconds in the Retry-After header.
Example Error:

```
HTTP/1.1 503 SERVICE UNAVAILABLE
Retry-After: 5
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 503,
  "detail": "The ward data is temporarily unavailable. Retry the request shortly."
}
```

Retrieving Groups (by groupId)

Endpoint: **$BASE_URL/scim/v3/Groups/{group_id}**
HTTP Operation: **GET**
Authorization: **Bearer Token**

A HTTP GET call to this endpoint can be used to return a single Group with its members. The members are the users who are linked to, or authorized for, the ward. Gerimedica support accounts are not part of the members and are never changed through SCIM.

Filtering, sorting and pagination are not supported.

Example Request

```
HTTP/1.1 GET $BASE_URL/scim/v3/Groups/linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12
Authorization: Bearer xxx
Content-Type: application/scim+json
```

Example Successful Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Location: $BASE_URL/scim/v3/Groups/linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Group"],
  "id": "linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12",
  "displayName": "Cardiology (Main building) - linked",
  "members": [
    {
      "value": "2819c223-7f76-453a-919d-413861904646",
      "display": "bjensen",
      "type": "User",
      "$ref": "$BASE_URL/scim/v3/users/2819c223-7f76-453a-919d-413861904646"
    },
    {
      "value": "5a1d6b3c-0e4f-4a7b-8c9d-1e2f3a4b5c6d",
      "display": "ssmith",
      "type": "User",
      "$ref": "$BASE_URL/scim/v3/users/5a1d6b3c-0e4f-4a7b-8c9d-1e2f3a4b5c6d"
    }
  ],
  "meta": {
    "resourceType": "Group",
    "location":
"$BASE_URL/scim/v3/Groups/linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12"
  }
}
```

Example Error Responses

Error Scenario: The caller attempts to get a Group which does not exist, or uses an id which is not a Group id.
Example Error:

```
HTTP/1.1 404 Not Found
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 404,
  "detail": "Group 'linked~unknown~1' was not found. A Group id has the form
'linked~<locationId>~<wardId>' or 'authorized~<locationId>~<wardId>'; GET /Groups
lists the valid ids."
}
```

Error Scenario: A request to this endpoint contains filtering options.
Example Error:

```
HTTP/1.1 400 BAD REQUEST
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidFilter",
  "detail": "Filtering is not supported for this operation."
}
```

Error Scenario: A request to this endpoint is made for an invalid organization.
Example Error:

```
HTTP/1.1 403 FORBIDDEN
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 403,
  "detail": "Operation is not permitted based on the supplied authorization."
}
```

Error Scenario: The location and ward data is temporarily unavailable.
Example Error:

```
HTTP/1.1 503 SERVICE UNAVAILABLE
Retry-After: 5
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 503,
  "detail": "The ward data is temporarily unavailable. Retry the request shortly."
}
```

Updating Group Members (PATCH)

Endpoint: **$BASE_URL/scim/v3/Groups/{group_id}**
HTTP Operation: **PATCH**
Authorization: **Bearer Token**

A HTTP PATCH call to this endpoint can be used to add members to, or remove members from, a Group. Only the members can be changed. The response contains the Group with the members as

accepted. The change itself is applied a few seconds later, see Asynchronous Processing. Fetch the Group before sending a PATCH.

- The operations ‘add’, ‘remove’ and ‘replace’ are supported. The operation name is not case sensitive.
- The path ‘members’ takes a list of members as its value. Each member has the userId in ‘value’. Other member attributes are ignored.
- A member can also be removed with the path ‘members[value eq "{userId}"]’. A remove with the path ‘members’ and no value removes all members.
- The path can be left out when the value is an object with the attribute ‘members’.
- Adding a user who is already a member, or removing a user who is not, has no effect and is not an error. A request can be repeated safely.
- Several operations in one request are applied in order. If one of them is invalid, nothing is changed.

Filtering, sorting and pagination are not supported.

Example Request

Add a member:

```
HTTP/1.1 PATCH
$BASE_URL/scim/v3/Groups/linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12
Authorization: Bearer xxx
Content-Type: application/scim+json
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "Operations": [
    {
      "op": "add",
      "path": "members",
      "value": [
        {
          "value": "5a1d6b3c-0e4f-4a7b-8c9d-1e2f3a4b5c6d"
        }
      ]
    }
  ]
}
```

Remove a member:

```
HTTP/1.1 PATCH
$BASE_URL/scim/v3/Groups/linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12
Authorization: Bearer xxx
Content-Type: application/scim+json
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "Operations": [
    {
      "op": "remove",
```

```
      "path": "members[value eq \"5a1d6b3c-0e4f-4a7b-8c9d-1e2f3a4b5c6d\"]"
    }
  ]
}
```

Example Successful Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Location: $BASE_URL/scim/v3/Groups/linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Group"],
  "id": "linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12",
  "displayName": "Cardiology (Main building) - linked",
  "members": [
    {
      "value": "2819c223-7f76-453a-919d-413861904646",
      "display": "bjensen",
      "type": "User",
      "$ref": "$BASE_URL/scim/v3/users/2819c223-7f76-453a-919d-413861904646"
    },
    {
      "value": "5a1d6b3c-0e4f-4a7b-8c9d-1e2f3a4b5c6d",
      "display": "ssmith",
      "type": "User",
      "$ref": "$BASE_URL/scim/v3/users/5a1d6b3c-0e4f-4a7b-8c9d-1e2f3a4b5c6d"
    }
  ],
  "meta": {
    "resourceType": "Group",
    "location":
"$BASE_URL/scim/v3/Groups/linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12"
  }
}
```

Example Error Responses

Error Scenario: The caller attempts to add a user which does not exist, is archived, or is not part of the organization.
Example Error:

```
HTTP/1.1 400 Bad Request
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidValue",
  "detail": "The value '5a1d6b3c-0e4f-4a7b-8c9d-1e2f3a4b5c6d' is not valid for the
'members' attribute."
}
```

Error Scenario: The caller attempts to change something other than the members, for example the displayName.
Example Error:

```
HTTP/1.1 400 Bad Request
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "mutability",
  "detail": "The 'displayName' attribute is readOnly. Only the 'members' of a Group
can be changed."
}
```

Error Scenario: The caller sends a request without operations.
Example Error:

```
HTTP/1.1 400 Bad Request
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidSyntax",
  "detail": "A PATCH request needs at least one entry in 'Operations'."
}
```

Error Scenario: The caller sends a path which is not supported.
Example Error:

```
HTTP/1.1 400 Bad Request
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidPath",
  "detail": "The path 'owners' is not supported. Use 'members' or 'members[value eq
\"<user id>\"]'."
}
```

Error Scenario: The caller attempts to change a Group which does not exist.
Example Error:

```
HTTP/1.1 404 Not Found
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 404,
  "detail": "Group 'linked~unknown~1' was not found. A Group id has the form
'linked~<locationId>~<wardId>' or 'authorized~<locationId>~<wardId>'; GET /Groups
lists the valid ids."
}
```

Error Scenario: The location and ward data is temporarily unavailable. Nothing was changed. Retry after the number of seconds in the Retry-After header.

Example Error:

```
HTTP/1.1 503 SERVICE UNAVAILABLE
Retry-After: 5
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 503,
  "detail": "The ward data is temporarily unavailable. Retry the request shortly."
}
```

Replacing Group Members (PUT)

Endpoint: **$BASE_URL/scim/v3/Groups/{group_id}**
HTTP Operation: **PUT**
Authorization: **Bearer Token**

A HTTP PUT call to this endpoint replaces all members of a Group. The members become exactly the users in the request. Users who are not in the request are removed from the Group. An empty list removes all members. The response and the processing are the same as for a PATCH, see Asynchronous Processing.

- The attribute ‘members’ is required. A request without it is refused.
- The attribute ‘displayName’ is ignored. The ‘id’ is optional and must be the id in the URL.
- Use PATCH unless the complete membership is always sent.

Filtering, sorting and pagination are not supported.

Example Request

```
HTTP/1.1 PUT $BASE_URL/scim/v3/Groups/linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12
Authorization: Bearer xxx
Content-Type: application/scim+json
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Group"],
  "id": "linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12",
  "members": [
    {
      "value": "2819c223-7f76-453a-919d-413861904646"
    },
    {
      "value": "5a1d6b3c-0e4f-4a7b-8c9d-1e2f3a4b5c6d"
    }
  ]
}
```

Example Successful Response

The same as the response of a PATCH.

Example Error Responses

Error Scenario: The caller sends a request without the ‘members’ attribute.
Example Error:

```
HTTP/1.1 400 Bad Request
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidSyntax",
  "detail": "A PUT needs the 'members' attribute. Send \"members\": [] to remove
every member."
}
```

Error Scenario: The caller sends an id in the body which is not the id in the URL.
Example Error:

```
HTTP/1.1 400 Bad Request
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "mutability",
  "detail": "The 'id' in the body ('linked~unknown~1') does not match the id in the
URL ('linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12')."
}
```

Error Scenario: The caller attempts to add a user which does not exist, is archived, or is not part of the organization.
Example Error:

```
HTTP/1.1 400 Bad Request
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": 400,
  "scimType": "invalidValue",
  "detail": "The value '5a1d6b3c-0e4f-4a7b-8c9d-1e2f3a4b5c6d' is not valid for the
'members' attribute."
}
```

Creating and Deleting Groups

A Group cannot be created or deleted through SCIM. Groups come from the wards in Ysis.

Error Scenario: The caller attempts to create a Group with a POST to $BASE_URL/scim/v3/Groups.
Example Error:

```
HTTP/1.1 405 Method Not Allowed
Allow: GET
```

Error Scenario: The caller attempts to delete a Group.
Example Error:

```
HTTP/1.1 405 Method Not Allowed
Allow: GET, PUT, PATCH
```

Groups of a User

In the v3 endpoints, a user has the read-only attribute ‘groups’. It lists the Groups the user is a direct member of. It is returned wherever a user is returned. A value sent on create or in a PUT is ignored, and a PATCH of this attribute is refused. Change the Group memberships of a user with the Group endpoints. The attribute is left out when the location and ward data is temporarily unavailable. Use ‘excludedAttributes=groups’ to leave it out.

Example

```
"groups": [
  {
    "value": "linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12",
    "display": "Cardiology (Main building) - linked",
    "type": "direct",
    "$ref": "$BASE_URL/scim/v3/Groups/linked~3f9c1e2a-5b7d-4c8e-9a10-2b3c4d5e6f70~12"
  }
]
```

Frequently Asked Questions

How do we assign a user to a location and a ward?
Add the user to the linked or authorized Group of the ward with a PATCH on the Group. See Assigning Users to Wards.

Is there a Group for a location?
No. Every Group is one ward of one location. The Group id and the Group name contain the location.

Does a successful response mean the user is assigned to the ward?
No. A 200 response means that the request is accepted. The change is applied a few seconds later. Retrieve the Group to confirm.

How long does it take before a change is applied?
A few seconds. A GET directly after a change can still show the previous members.

Can we create, rename or delete locations and wards through SCIM?
No. Locations and wards are managed in Ysis. SCIM decides which users are in the Groups of a ward.

How do we get the location and ward ids?
Call GET /Groups. The id of each Group contains the location id and the ward id, and the displayName contains the ward and location names. Use this list to map the wards of your system to the wards in Ysis.

Can we assign a ward when we create the user?
No. Create the user first, then add the user to the Groups. The ‘groups’ attribute of a user is read-only.

Should we use PATCH or PUT?
Use PATCH to add or remove individual members. A PUT replaces all members: everyone who is not in the list is removed.

Is it safe to send the same request twice?
Yes. Adding a member who is already in the Group, or removing a user who is not, has no effect and gives no error.

What is the difference between a linked and an authorized Group?
Linked: the user works on the ward. Authorized: the user has access to the ward. Adding a user to an authorized Group turns on the restriction per ward for that user, as in the manual flow.

The response was 200 but the user is not in the Group. Why?
Wait a few seconds and retrieve the Group again. If the user is still missing, check that the user exists, is not archived and does not have the discipline ADMINISTRATIVE. Ysis can refuse a change which breaks a ward rule. Contact support if the problem remains.

What happens to the ward assignments when a user is deleted?
The user is archived and removed from all wards. This is processed asynchronously.

How do we re-activate a deleted user?
Send a create request with the same userName, ysisInitials and discipline as the archived user. The user keeps its userId. The ward assignments are not restored.

How do we find an archived user?
Use the list users endpoint with a filter and ‘includeArchived=true’. Archived users have ‘active’ set to false.

Why do we get 409 Conflict when we create a user?
The userName or the ysisInitials is already used in the organization. If it belongs to an archived user, the detail of the error explains how to re-activate that user, or how to create a new user instead.

Why does the list of users return only 100 users?
Without a filter the list is paged. Use ‘startIndex’ and ‘count’ to select the pages.

Can we change a user with a PATCH?
Yes. A PATCH changes only the attributes in the request. A PUT replaces the whole user. See Updating Users (PATCH).

How do we leave the groups out of a user?
Add ‘excludedAttributes=groups’ to the request.

What does 503 on the Group endpoints mean?
The location and ward data is temporarily unavailable. Nothing was changed. Retry after the number of seconds in the ‘Retry-After’ header.

Is the list of Groups paged?
No. All Groups of the organization are returned.

Which API version has the Groups?
Version 3 (scim/v3).

Who can we contact with questions?
Gerimedica support via support@gerimedica.nl.